Semantic-Oriented Cross-Lingual Ontology Mapping

A thesis submitted to the

University of Dublin, Trinity College

for the degree of

Doctor of Philosophy

Bo Fu

Knowledge and Data Engineering Group

Intelligent Systems Lab

School of Computer Science and Statistics

Trinity College Dublin

Ireland

2011

DECLARATION

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work. I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Bo Fu

November 2011

ACKNOWLEDGEMENTS

I would like to deeply thank my supervisor Dr. Declan O'Sullivan for his guidance, encouragement and support over the years. My sincere thanks also to my assistant supervisor Dr. Rob Brennan for his valuable suggestions and great patience. I would also like to express my gratitude to Prof. Vincent Wade for providing me with the necessary support during this research. A special thank you to Dr. Owen Conlan and Dr. Dave Lewis for their insightful feedback. A warm thank you to all members of KDEG from whom I have learned much. I would like to especially thank Stephen, Melike, Alex, Shay, Zhou, Wei, Paul, Colm, Brian, Hendrik and Vicky for their collaboration during this research. A huge thank you to Trinity College Dublin and the Centre of Next Generation Localisation for funding this research.

My greatest appreciation to all my wonderful friends who have shown much love and support. I am most grateful for the unconditional support and unwavering belief from Ben and my parents. To my grandparents, this thesis is dedicated to them.

ABSTRACT

Ontologies support knowledge discovery, sharing and reuse among people and enable semantic interoperability between computer-based systems. To establish correspondences between knowledge concepts represented in ontologies, ontology mapping is at the heart of dealing with heterogeneity on the semantic web. A great deal of effort has focused on the matching of ontologies that are written in the same natural language and various tools have been developed to facilitate this monolingual ontology matching process. However, as knowledge and knowledge representations are not restricted to the usage of a single natural language, to make use of knowledge bases across natural language barriers, matching tools and techniques must be able to work with ontologies that are written in heterogeneous natural languages. This research identifies key challenges, tools and techniques to support the process of cross-lingual ontology mapping between independent ontologies that are written in diverse natural languages. One approach to cross-lingual ontology mapping (CLOM): the translation-based approach, is to use translation techniques to convert a cross-lingual mapping problem into a monolingual mapping problem which can then be solved via existing monolingual matching tools. However, noise can be introduced during the translation process which leads to poor mapping quality in the subsequent monolingual matching step. This thesis aims to address this challenge faced by translation-based approach to cross-lingual ontology mapping by proposing the concept of appropriate ontology label translation (AOLT). Appropriate translations in the context of cross-lingual ontology mapping are those translations that are most likely to maximise the success of the subsequent monolingual ontology matching step. In particular, this thesis presents two realisations of the AOLT concept, which have been integrated in two Semantic-Oriented Cross-lingual Ontology Mapping systems: SOCOM and SOCOM++. It is shown through the evaluations of SOCOM and SOCOM++ that the proposed AOLT concept is effective at improving CLOM quality compared to the baseline system. A major contribution of this thesis is the AOLT concept, its demonstration and evaluation. The proposed AOLT concept distinguishes translations that take place for the purpose of cross-lingual ontology mapping and those that take place for the purpose of localisation. This AOLT concept is the first attempt that aims to improve mapping quality in translation-based cross-lingual ontology mapping systems.

TABLE OF CONTENTS

DECLARATION i
ACKNOWLEDGEMENTS ii
ABSTRACT iii
TABLE OF CONTENTS iv
TABLE OF FIGURES vii
TABLE OF TABLES ix
ABBREVIATIONS x
1 INTRODUCTION 1
1.1. Chapter Overview 1
1.2. Motivation 1
1.3. Research Question 4
1.4. Objectives and Goals 5
1.5. Technical Approach 6
1.6. Contribution 8
1.7. A Glossary of Terminologies 9
1.8. Thesis Overview 10
2 BACKGROUND AND STATE OF THE ART 12
2.1. Chapter overview 12
2.2. The Ontology Mapping Problem 12
2.3. Ontologies and Multilinguality 14
2.4. Cross-Lingual Ontology Mapping 16
2.4.1. Categories of CLOM Approaches 17
2.4.2. Translations in CLOM vs. Translations in Ontology Localisation 22
2.5. Monolingual Ontology Mapping 24
2.5.1. Categories of MOM Techniques 24
2.5.2. The Alignment API 30
2.6. Machine Translation 31
2.7. Evaluation Metrics for Ontology Mapping 33
2.7.1. A Tutorial on Evaluation Metrics 34
2.7.1.1. Precision, Recall and F-Measure 34
2.7.1.2. Paired T-Test 36
2.7.1.3. Mean and Standard Deviation 38
2.7.2. Evaluation Metrics used in This Thesis 38
2.8. Summary 43
3 THE CHALLENGE OF TRANSLATION IN CROSS-LINGUAL ONTOLOGY MAPPING 44
3.1. Chapter Overview 44
3.2. Experimental Motivation 45
3.3. The Baseline Approach 45
3.3.1. Architecture Overview 45
3.3.2. Implementation 47
3.4. Experiments 53
3.4.1. Experiment One 55
3.4.1.1. Experimental Setup 55
3.4.1.2. Findings and Analysis 57
3.4.2. Experiment Two 59
3.4.2.1. Experimental Setup 59
3.4.2.2. Findings and Analysis 61

3.4.3. Conclusions ... 62
3.5. Summary ... 65
4 PROTOTYPE ONE: SOCOM ... 66
4.1. Chapter Overview ... 66
4.2. The Concept of Appropriate Ontology Label Translation ... 67
4.3. SOCOM Design ... 70
4.4. SOCOM Implementation ... 75
4.5. SOCOM Evaluation ... 81
4.5.1. Experiment One ... 83
4.5.1.1. Experimental Setup ... 83
4.5.1.2. Findings and Analysis ... 87
4.5.2. Experiment Two ... 91
4.5.2.1. Experimental Setup ... 92
4.5.2.2. Findings and Analysis ... 93
4.5.3. Conclusions ... 96
4.6. Case Study ... 97
4.6.1. Objectives and Scope of the Case Study ... 98
4.6.2. Background of the Case Study ... 99
4.6.3. Design, Implementation and Execution ... 101
4.6.4. Significance of the Case Study ... 106
4.7. Summary ... 108
5 PROTOTYPE TWO: SOCOM++ ... 109
5.1. Chapter Overview ... 109
5.2. SOCOM++ Design ... 110
5.3. SOCOM++ Implementation ... 114
5.4. SOCOM++ Evaluation ... 128
5.4.1. Overview of Trials ... 129
5.4.2. Three Trials to adjust Ontology Semantics ... 134
5.4.2.1. Trial One - adjust Execution Constraint ... 134
5.4.2.1.1. Trial Setup ... 134
5.4.2.1.2. Findings and Analysis ... 136
5.4.2.2. Trial Two - adjust Resource Constraint ... 140
5.4.2.2.1. Trial Setup ... 140
5.4.2.2.2. Findings and Analysis ... 141
5.4.2.3. Trial Three - adjust Embedded Semantics ... 145
5.4.2.3.1. Trial Setup ... 145
5.4.2.3.2. Findings and Analysis ... 146
5.4.3. Three Trials to execute a Second Iteration of the AOLT Process 149
5.4.3.1. Trial Four - adjust Task Intent: Optimising Correctness ... 149
5.4.3.1.1. Trial Setup ... 150
5.4.3.1.2. Findings and Analysis ... 152
5.4.3.2. Trial Five - Optimising Completeness ... 155
5.4.3.2.1. Trial Setup ... 155
5.4.3.2.2. Findings and Analysis ... 156
5.4.3.3. Trial Six - adjust Pseudo Feedback ... 160
5.4.3.3.1. Trial Setup ... 160
5.4.3.3.2. Findings and Analysis ... 161
5.4.4. Conclusions arising out of the Six Trials ... 164
5.4.5. Scalability Tests ... 170
5.4.5.1. Tests Setup ... 170
5.4.5.2. Findings and Analysis ... 172
5.5. Summary ... 176

6 CONCLUSIONS 177
6.1. Chapter Overview 177
6.2. Objectives 177
6.3. Contributions 182
6.4. Future Work 185
6.5. Final Remarks 186
BIBLIOGRAPHY 188
APPENDICES 206
APPENDIX A. DVD CONTENT A-1
APPENDIX B. MONOLINGUAL MATCHING TOOLS B-1
APPENDIX C. CODE SNIPPETS C-1
C.1. Appendix Overview C-1
C.2. The Baseline System Code Snippets C-1
C.3. Prototype One: SOCOM Code Snippets C-3
C.4. Prototype Two: SOCOM++ Code Snippets C-8
APPENDIX D. DOCUMENT TYPE DEFINITIONS D-1
D.1. Appendix Overview D-1
D.2. DTDs in Prototype One: SOCOM D-1
D.3. DTDs in Prototype Two: SOCOM++ D-1
APPENDIX E. CONFIDENCE LEVEL EVALUATION PLOTS E-1
E.1. Trial One E-1
E.2. Trial Two E-2
E.3. Trial Three E-2
E.4. Trial Four E-3
E.5. Trial Five E-4
E.6. Trial Six E-5

TABLE OF FIGURES

Figure 1-1. Natural Language Content as Resource Identifiers 2
Figure 2-1. Euzenat & Shvaiko's Classification of Matching Approaches [Euzenat & Shvaiko, 2007 p.65] 25
Figure 2-2. Ehrig's Similarity Layers [Ehrig, 2007 p.27] 28
Figure 2-3. An Example Output from the Alignment API 31
Figure 2-4. Precision & Recall in Mapping Evaluation 35
Figure 2-5. Two Tailed and One Tailed Paired T-Test [Weiss, 2010 p.481] 37
Figure 2-6. Paired T-Test for M_1 and M_2 42
Figure 3-1. An Architecture of the Baseline Approach to CLOM 46
Figure 3-2. An Example of Ontology Rendition 47
Figure 3-3. An Implementation of the Baseline Approach to CLOM - Deployment Diagram 48
Figure 3-4. Ontology Rendition - UML Sequence Diagram 50
Figure 3-5. Partial Views of the SWRC and the ISWC Ontology in Protégé 54
Figure 3-6. Experiment One Overview 56
Figure 3-7. Experiment One Evaluation Results 58
Figure 3-8. Experiment Two Overview 60
Figure 3-9. Experiment Two Evaluation Results 61
Figure 4-1. Examples of AOLT in the Context of CLOM 68
Figure 4-2. Workflow in SOCOM 71
Figure 4-3. The AOLT Process in SOCOM 74
Figure 4-4. An Example of the Translation Repository 77
Figure 4-5. An Example of the Lexicon Repository 78
Figure 4-6. An Example Output from Term Extraction 78
Figure 4-7. UML Class Diagram of SOCOM 81
Figure 4-8. The Creation Process of the CSWRC Ontology 85
Figure 4-9. Experiment One Overview 86
Figure 4-10. Experiment One Precision, Recall and F-Measure Results 88
Figure 4-11. Evaluation on Confidence Levels - Experiment One 90
Figure 4-12. Experiment Two Overview 93
Figure 4-13. Experiment Two Precision, Recall and F-Measure Results 94
Figure 4-14. Evaluation on Confidence Levels - Experiment Two 95
Figure 4-15. CLOM-Enabled CLIR 100
Figure 4-16. The Ontologies in the Case Study 102
Figure 4-17. SOCOM in CLIR 104
Figure 4-18. Screenshots from the ARCHING System 106
Figure 5-1. Prototype Two: SOCOM++ Design Diagram 111
Figure 5-2. The Configuration File in SOCOM++ 114
Figure 5-3. Implementation Diagram of SOCOM++ 117
Figure 5-4. An Example of O_1 Analysis 119
Figure 5-5. An Example of O_2 Analysis 120
Figure 5-6. Class Diagram of Ontology Semantic Analysis in SOCOM++ 121
Figure 5-7. UML Class Diagram of Ontology Rendition in SOCOM++ 123
Figure 5-8. An Example of AOLT Record 124
Figure 5-9. An Example of Final AOLT Results 127
Figure 5-10. Class Diagram of Mapping Generation 127
Figure 5-11. Experimental Setup of Trial One to Six 133
Figure 5-12. Precision, Recall and F-Measure found in Trial One 137

Figure 5-13. Precision, Recall, F-Measure in SOCOM++ Trial Two 142
Figure 5-14. Precision, Recall, F-Measure in Trial Three 147
Figure 5-15. An Example Output from Task Intent Analysis - Optimising Correctness 150
Figure 5-16. Precision, Recall, F-Measure in Trial Four 153
Figure 5-17. An Example Output from Task Intent Analysis - Optimising Completeness 156
Figure 5-18. Precision, Recall, F-Measure in Trial Five 157
Figure 5-19. An Example Ouput from Pseudo Feedback Analysis 161
Figure 5-20. Precision, Recall, F-Measure in Trial Six 162
Figure C-1. OntModel Creation C-1
Figure C-2. OntClass Creation in O_1' C-2
Figure C-3. Datatype Property Creation in O_1' C-2
Figure C-4. Creating Matches with the Alignment API C-3
Figure C-5. Iterating through the Classes in an Ontology C-3
Figure C-6. Class Semantic Surrounding Generation C-4
Figure C-7. Generating and Storing Candidate Translations C-4
Figure C-8. Synonym Generation for an Individual in O_2 .... C-5
Figure C-9. Keyword Generation for a Candidate Translation C-6
Figure C-10. String Comparison C-7
Figure C-11. Semantic Surrounding Comparison C-7
Figure C-12. Collision Resolution in SOCOM C-8
Figure C-13. English Synonym Generation via the Big Huge Thesaurus API C-9
Figure C-14. French Synonym Generation via synonyms-fr.com C-9
Figure C-15. Property Validation C-10
Figure C-16. XQuery Example C-10
Figure C-17. XPathQuery Example C-11
Figure D-1.Translation Repository DTD D-1
Figure D-2. Lexicon Repository DTD..D-1
Figure D-3. Configuration File DTD...D-1
Figure D-4. Source Semantic DTD...D-2
Figure D-5. Target Semantic DTD..D-2
Figure D-6. AOLT Record DTD...D-3
Figure D-7. AOLT Selection DTD...D-3
Figure D-8. Task Intent DTD..D-3
Figure D-9. Pseudo Feedback DTD...D-3
Figure E-1. Confidence Evaluation in Trial One..E-1
Figure E-2. Confidence Evaluation in Trial Two..E-2
Figure E-3. Confidence Evaluation in Trial Three..E-3
Figure E-4. Confidence Evaluation in Trial Four..E-4
Figure E-5. Confidence Evaluation in Trial Five..E-5
Figure E-6. Confidence Evaluation in Trial Six..E-5

TABLE OF TABLES

Table 2-1. An Overview of Translation-Based CLOM Approaches 19
Table 2-2. Matching Algorithms in the Alignment API 31
Table 2-3. Examples of Precision, Recall and F-Measure 35
Table 2-4. Paired T-Test on F-Measure 42
Table 3-1. Translation Noise during Ontology Rendition 63
Table 4-1. Collision Resolution in SOCOM 80
Table 4-2. Experiment One Confidence Data 91
Table 4-3. Experiment Two Confidence Data 95
Table 4-4. Key Findings in Experiment One and Two 96
Table 5-1. Types of Candidate AOLT Results 125
Table 5-2. A Summary of SOCOM++ Trial Configurations 130
Table 5-3. Collision Resolution in SOCOM++ Trial One 136
Table 5-4. Confidence Data from Trial One 139
Table 5-5. Key Findings of Baseline, SOCOM and SOCOM++ Trial One 140
Table 5-6. Collision Resolution in Trial Two 141
Table 5-7. Confidence Data from Trial Two 143
Table 5-8. Key Findings of Baseline, SOCOM++ Trial One and Two 144
Table 5-9. Collision Resolution in Trial Three 146
Table 5-10. Confidence Data from Trial Three 148
Table 5-11. Key Findings of Baseline, SOCOM++ Trial One, Two and Three 149
Table 5-12. Collisions Resolution in Trial Four 152
Table 5-13. Confidence Data from Trial Four 154
Table 5-14. Key Findings of Baseline, SOCOM++ Trial One and Four 155
Table 5-15. Confidence Data from Trial Five 159
Table 5-16. Key Findings of Baseline, SOCOM++ Trial One and Five 159
Table 5-17. Confidence Data from Trial Six 163
Table 5-18. Key Findings of Baseline, SOCOM++ Trial One and Six 163
Table 5-19. An Overview of the SOCOM++ Trials in Experiment One 165
Table 5-20. An Overview of the SOCOM++ Trials in Experiment Two 165
Table 5-21. Rankings of CLOM Systems in Experiment One 167
Table 5-22. Rankings of CLOM Systems in Experiment Two 167
Table 5-23. An Overview of Ranks achieved in Experiment One and Two 169
Table 5-24. Ontologies used in the Scalability Tests 171
Table 5-25. Scalability Tests' Results 173

ABBREVIATIONS

AOLT	Appropriate Ontology Label Translation
API	Application Programming Interface
BHT	The Big Huge Thesaurus
CLOM	Cross-Lingual Ontology Mapping
CLIR	Cross-Lingual Information Retrieval
CNGL	Centre for Next Generation Localisation
CPU	Central Processing Unit
DTD	Document Type Definition
DVD	Digital Versatile Disc
FAO	Food and Agriculture Organization
HTML	HyperText Markup Language
IR	Information Retrieval
JRE	Java Runtime Environment
JVM	Java Virtual Machine
OWL	Web Ontology Language
PDF	Portable Document Format
RDF	Resource Description Framework
MOM	Monolingual Ontology Mapping
MT	Machine Translation
UML	Unified Modelling Language
URI	Unique Resource Identifier
SOCOM	Semantic-Oriented Cross-Lingual Ontology Mapping
SWRC	Semantic Web Research Community
WSD	Word Sense Disambiguation
XML	Extensible Markup Language

1 INTRODUCTION

1.1. Chapter Overview

This chapter presents an overview of this thesis. In particular, the motivation of this research is discussed in section 1.2. The research question addressed by this thesis is presented in section 1.3. A list of objectives and goals derived from this research question are discussed in section 1.4. The technical approach undertaken for this research is presented in section 1.5, followed by a discussion of the contributions in section 1.6. A glossary of terminologies used in this thesis is included in section 1.7. Finally, section 1.8 presents an overview of the remaining chapters of this thesis.

1.2. Motivation

Berners-Lee et al. define the semantic web as "an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [Berners-Lee et al., 2001]. Comparing to the current web, where information is presented for humans to read and understand, on the semantic web, information is encoded in semantics that can be read and understood by machines. Ontologies, as specifications of conceptualisations [Gruber, 1993], are recognised as a "basic component of the semantic web" in [Berners-Lee et al., 2001] and have been widely used in knowledge management in recent years [Jurisica et al., 2004].

One approach to ontology construction, is to use language neutral identifiers to label concepts [Nirenburg & Raskin, 2001], whereby ontological resources are natural language independent. However, this view is debatable. For instance, as Bateman points out "the path towards viable ontologies is one that is irreconcilably connected to natural language" [Bateman, 1993]. Also in practice, natural language labels are commonly used in ontological resource identifiers as seen in [Noy & McGuinness, 2001; Horridge et al., 2004]. For instance, figure 1-1 presents a snippet of the pizza ontology[1] used in the Protégé tutorial [Horridge et al., 2004] where a class is labelled with *CheeseTopping* in natural language. As a result of the use of natural languages in resource naming during ontology development, ontologies that are labelled in diverse natural languages are increasingly evident. For instance, at the time of this writing, the OntoSelect Ontology Library[2] reports over 25% of 1530 ontologies indexed are written in natural languages other than English.

```
<owl:Class rdf:about="#CheeseTopping">
   <rdfs:label xml:lang="pt">CoberturaDeQueijo</rdfs:label>
   <rdfs:subClassOf>
   <owl:Class rdf:about="#PizzaTopping"/>
   </rdfs:subClassOf>
</owl:Class>
```

Figure 1-1. Natural Language Content as Resource Identifiers

Given ontologies that are likely to be authored by different actors using different terminologies, structures and natural languages, ontology mapping - the process of generating correspondences between ontological resources [Euzenat & Shvaiko, 2007] - has emerged as a way to achieve semantic interoperability. To date, research in the field of ontology mapping has largely focused on dealing with ontologies that are labelled in the same natural language[3], little research has focused on providing assistance and support in mapping scenarios where the ontologies involved are labelled in different natural languages. The issue with current matching techniques is that they often rely on lexical comparisons made between resource identifiers, which limits their deployment to ontologies labelled in the same natural language or at least in comparable natural languages[4]. For example, a match may be established between a class <owl:Class rdf:about="#Cheese"> in the source ontology and a class <owl:Class rdf:about="#cheese"> in the target ontology (i.e. both ontologies are in English). However, when lexical comparison is not possible between two natural languages (e.g. English and Chinese from different language families), a match to the class <owl:Class rdf:about="#奶酪"> in the target ontology would be neglected given

1 http://www.co-ode.org/ontologies/pizza/2007/02/12/pizza.owl
2 http://olp.dfki.de/ontoselect?wicket:bookmarkablePage=:de.dfki.ontoselect.Statistics
3 A survey of monolingual matching tools is presented in [Euzenat & Shvaiko, 2007 chapter 6], example tools can be found in appendix B and a discussion can be found in chapter 2, section 2.5.
4 An example of comparable natural languages can be English and French, which both belong to the same Germanic language family. Another example is Italian and German, though the former belongs to the Romance language family and the latter belongs to the Germanic language family, however, they are both alphabetic letter-based hence have comparable graphemes that can be analysed using string comparison techniques such as edit distance. An example of natural languages that are not comparable in this context can be Chinese and English, where edit distance is not applicable since the graphemes in the former is logogram-based and the graphemes in the latter is alphabetic letter-based.

monolingual matching tools. Even though multilingual support can be provided to ontologies via language tagging to assist monolingual matching tools without changing the natural language segments in the resource identifiers, this form of assistance may not always be available to every mapping scenario. For example, in figure 1-1, `<rdfs:label xml:lang="pt">CoberturaDeQueijo</rdfs:label>` tags the label of the *CheeseTopping* class with *CoberturaDeQueijo* in Portuguese. Assuming the other ontology is in Portuguese, the `<rdfs:label>` element content may then be used by monolingual matching tools instead of the class identifier to generate matches in this case. However, such an approach requires all the resources in a given ontology to be tagged with target natural language content, which may be a difficult requirement. To the best of this author's knowledge, mapping tools that use multilingual resource tagging are not yet available.

Given the limitations of existing matching tools that focus on mostly monolingual matching processes, there is a pressing need for the development of matching techniques that can work with ontologies in different natural languages. One way to enable semantic interoperability between ontologies in different natural languages is by means of cross-lingual ontology mapping. In this thesis, *cross-lingual ontology mapping (CLOM) refers to the process of establishing relationships among ontological resources from two or more independent ontologies where each ontology is labelled in a different natural language*.

A popular approach [Zhang et al., 2008; Bouma, 2010; Wang et al., 2009; Trojahn, 2010] to achieve CLOM is to use translation techniques with the goal of converting a cross-lingual mapping problem into a monolingual mapping problem which can then be solved by state of the art monolingual ontology matching (MOM) tools (for a detailed discussion, see chapter 2, section 2.5). This translation-based CLOM process can be summarised as follows: given ontologies O_1 and O_2 that are labelled in different natural languages, the labels of one of them, for example, O_1, are first translated into the natural language used by O_2. As both ontologies are now labelled in the same natural language, the mappings between them can then be created by simply applying monolingual ontology matching techniques. The intermediary step involving the translation of ontological resource labels is often achieved by using machine translation (MT) techniques. Various techniques [Clark et al., 2010] such as statistical MT and rule-based MT have been developed, which aim to improve the quality of translation via word sense disambiguation [Navigli, 2009]. More importantly, MT tools are intended to generate the most accurate translations in the linguistic sense, which is not necessarily a requirement in CLOM. To achieve CLOM, translation is merely an intermediate stage to the actual goal which is generating mappings between ontological resources. Consequently, translating the labels of the source ontology is not solely concerned with finding linguistic equivalents in the target natural language, but also finding translations that can lead to the discovery of quality mappings.

There can be various ways to express the same or a very similar concept in many natural languages. A simple example of this is: the term *Ph.D. candidate* and the term *doctoral student* can both describe someone who is pursuing an academic degree of Doctor of Philosophy. Envision this in the context of cross-lingual ontology mapping, assuming the target ontology is labelled in English and the source ontology is labelled in a natural language other than English. For an ontological resource in the source ontology, its English translation can be *Ph.D. candidate*, it can also be *doctoral student*. But which one of these candidate translations is more appropriate in the given mapping scenario? To answer this question, we would ideally like to know which candidate translation will lead to a successful mapping given that an equivalent semantic resource is presented in the target ontology. This translation selection process differs from traditional word sense disambiguation (WSD) in the context of natural language processing, as WSD is "the association of a given word in a text or discourse with a definition or meaning (sense) which is distinguishable from other meanings potentially attributable to that word" [Ide & Véronis, 1998]. In the context of translation-based CLOM, the outcome of the mapping process is conditioned on the translations selected for the given ontology resources. In order to generate quality mapping results, translations must be selected appropriately. This idea of using *appropriately selected translations* to assist MOM tools in the CLOM process is the focus of this thesis, and is validated in a series of experiments.

1.3. Research Question

This research investigates the extent to which machine translation (MT) and monolingual ontology mapping (MOM) techniques can be incorporated to support the generation of quality mapping results in the process of cross-lingual ontology mapping (CLOM).

As introduced earlier, CLOM refers to the process of establishing relationships among ontological resources from two or more independent ontologies where each ontology is labelled in a different natural language. To measure mapping quality, evaluation metrics such as precision, recall, f-measure, paired t-test, mean and standard deviation are used. Details of these metrics are presented in chapter 2, section 2.7.

In computer science, Gruber's definition of an ontology as "explicit specification of a conceptualisation" [Gruber, 1993] is widely accepted. Examples of ontologies include folksonomies, lexicon databases, directories, thesauri and formal ontologies, as discussed in [Euzenat & Shvaiko, 2007, p.29]. The focus of this Ph.D. is formally defined ontologies that follow the Resource Description Framework[5] (RDF) schema or the Web Ontology Language[6] (OWL) specification. The focus of the CLOM process presented in this thesis is the generation of correspondences between ontological resources in formally defined multilingual ontologies. In this thesis, *multilingual ontologies* refer to two (i.e. a pair of) or more (i.e. a group of) independent ontologies containing resources that do not share the use of a common natural language. It does not refer to ontologies that contain resources with multiple natural languages at once (such as the bilingual thesaurus presented in [Shimoji et al., 2008]). In addition, these ontologies have not been linguistically enriched (e.g. the ontological resources are associated with linguistic information as presented in [Pazienza & Stellato, 2006a]), nor do they have multiple multilingual natural language content associated with the same ontological resource (such as the example shown in figure 1-1).

1.4. Objectives and Goals

To address the research question discussed in section 1.3, the following objectives have been derived:

- Conduct reviews on the state of the art in cross-lingual ontology mapping, machine translation, monolingual ontology mapping and current approaches to the evaluation of mapping results.
- Design and develop a process specifically suited for translations carried out for the purpose of CLOM and implement a set of tools to support this translation process in order to achieve CLOM results via MOM techniques.

[5] http://www.w3.org/TR/rdf-schema
[6] http://www.w3.org/TR/owl-features

- Evaluate the quality of the mappings generated using the set of tools in CLOM scenarios and demonstrate the use of the set of tools in a real-world application.

1.5. Technical Approach

A state of the art review (discussed in chapter 2) is conducted first in the field of cross-lingual ontology mapping, and a popular approach is identified. This approach to CLOM uses machine translation as a means to turn a cross-lingual mapping problem into a monolingual mapping problem which can then be solved by monolingual ontology matching tools. Surveys are thus carried out on the state of the art in MT and MOM, where appropriate tools to assist the CLOM process are identified. A baseline system (discussed in chapter 3) is implemented based on this identified approach, and evaluated in a set of experiments involving ontologies labelled in Chinese and English. The findings from the experiments suggest that translation noise can have a negative impact on the subsequent monolingual matching step, which can lead to poor mapping quality as a result.

Based on this finding, the concept of appropriate ontology label translation (AOLT) was developed to facilitate the translations carried out in the context of CLOM (discussed in chapter 4). To realise the proposed AOLT concept, the AOLT process is then developed. The outcome from the AOLT process is referred to in this thesis as the AOLT results. A definition of the AOLT concept is presented in chapter 4, section 4.2. The AOLT concept aims to select appropriate ontology label translations, where the appropriateness of a translation is determined by its likelihood to lead to a successful mapping (given that such a mapping exists in the given CLOM scenario). The goal is to select translations from a pool of candidate translations that are most likely to maximise the matching ability of the subsequent monolingual matching techniques. To demonstrate the AOLT concept in the process of achieving CLOM, the Semantic-Oriented Cross-lingual Ontology Mapping (SOCOM) system is developed that generates CLOM results through the use of the AOLT process. Though there may be other ways to realise the AOLT concept (discussed in section 4.2), in this thesis, since the AOLT concept is realised through analysing the semantics (i.e. using translations and synonyms to illustrate the *meaning* of ontology labels, as well as analysing the semantic surroundings of nodes based on the ontological *graph*) of the ontologies involved in the CLOM scenario, the prototypes (SOCOM and SOCOM++) presented in this thesis are thus considered as *semantic-oriented* cross-lingual ontology mapping systems. The goal of the SOCOM system is to support the use of MT and MOM techniques in CLOM processes by applying the AOLT process. SOCOM is evaluated in a set of CLOM experiments involving ontologies labelled in Chinese, English and French. The findings showed an improvement in matching quality when the AOLT process is applied in comparison to the baseline system. The applicability of SOCOM is also demonstrated in an ontology-based, adaptive customer support system case study (discussed in chapter 4, section 4.6). This case study aims to provide users with relevant information in more than one natural language. The application retrieves documents within the domain of Symantec's home security product: Norton 360[7]. The underlying ontologies used by this application are labelled in English and German, and SOCOM is applied to achieve a composed presentation of the knowledge base through CLOM results. This case study aims to showcase the feasibility of SOCOM in a real-world application.

Motivated by the positive findings from the initial CLOM prototype: SOCOM, an improved second prototype: SOCOM++ is designed and implemented (discussed in chapter 5). SOCOM++ implements a more sophisticated AOLT process, which takes configurable inputs during the AOLT process and in turn influences the CLOM outcome. The implementation investigates whether SOCOM++ can be adjusted to suit specific needs of a given CLOM setting in the generation of high quality mappings. A set of experiments have been carried out to evaluate this improved prototype with the same ontology pairs used in the SOCOM evaluations. The flexibility of the AOLT process was demonstrated in the experiments, and the findings show that a range of quality levels were achieved using varied configurations of SOCOM++. The scalability aspect (in terms of execution time) of SOCOM++ was also investigated in CLOM experiments involving large ontology pairs (taken from the OAEI 2008 contest) with thousands of ontological resources labelled in English and Japanese. The experiment results showed increased processing time with increased workload and increased complexity of the AOLT configuration. The benefit of the AOLT process and its ability to scale is demonstrated through SOCOM++'s ability to work with large ontologies.

The approach undertaken by this research when evaluating CLOM results applies metrics (discussed in chapter 2, section 2.7) that are currently used in the state of the art

[7] http://us.norton.com/360

for evaluating matches generated by MOM systems. These metrics are suitable for the CLOM result evaluation as the goal of the evaluation remains unchanged (whether it is a monolingual or multilingual mapping environment): measuring how correct and complete are a set of matches against a gold standard. In particular, precision is used to evaluate the correctness and recall is used to evaluate the completeness of a set of matches. F-measure is used to evaluate the overall quality of a set of matches as it accounts both precision and recall. Means and standard deviations are used to evaluate the confidence levels of the matches generated. In addition, paired t-tests are carried out to validate the statistical significance of the findings in each experiment. Finally, the scalability test (discussed in chapter 5, section 5.4.5) measures the execution time required in different CLOM scenarios (e.g. increased workload with larger ontologies to process). In particular, the execution time of a simpler (e.g. less inputs into the AOLT process) and a more complex (e.g. more inputs into the AOLT process) configuration of SOCOM++ are investigated.

1.6. Contribution

A major scientific contribution of this thesis is the concept of applying appropriate ontology label translations to improve the quality of results arising from a cross-lingual ontology mapping process. An appropriate ontology label translation (AOLT) in the context of cross-lingual ontology mapping is one that is most likely to maximize the success of the subsequent monolingual ontology mapping step. This is a novel concept in achieving ontology label translations that are carried out for the purpose of cross-lingual ontology mapping. The proposed AOLT concept is successfully demonstrated and evaluated in this thesis. It is shown through the evaluations that appropriate ontology label translations are effective at improving cross-lingual ontology mapping quality. In addition, this thesis differentiates translation noise that occurs in the context of localisation and those that occur in the context of cross-lingual ontology mapping. Reducing translation noise in the context of localisation is centred on generating translations that are the same with/close to human translations, whereas reducing translation noise in the context of cross-lingual ontology mapping is centred on generating translations that lead to quality cross-lingual ontology mapping results via monolingual ontology mapping techniques. The impact of ontology label translations on the final mapping quality is examined in this thesis, which has not yet been investigated previously in the state of the art of cross-lingual ontology mapping.

A minor contribution of this thesis is the AOLT processes that have been implemented in two cross-lingual ontology mapping systems (SOCOM and SOCOM++) which realise the proposed AOLT concept. Although there may be other ways to realise the AOLT concept, the AOLT processes presented in this thesis are not an exhaustive list but rather example implementations. These AOLT processes are demonstrated and evaluated through a series of cross-lingual ontology mapping experiments. It is shown through the evaluations that the AOLT process is an effective procedure at improving mapping quality in cross-lingual ontology mapping scenarios.

Five peer-reviewed scientific publications have derived from this research, including two full research papers at the 8th Extended Semantic Web Conference (ESWC 2011) and the 4th Asian Semantic Web Conference (ASWC 2009), one research poster at the 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2010), two workshop papers at the 1st workshop on the Multilingual Semantic Web (MSW 2010) which was collocated at the 19th International World Wide Web conference (WWW 2010) and the workshop on Matching and Meaning (2009). Details of these publications can be found in chapter 6 (section 6.3).

1.7. A Glossary of Terminologies

For clarification purposes, this section presents a short glossary of terminologies used in this thesis.

- **AOLT concept** refers to the abstract concept of appropriate ontology label translations whereby appropriateness is determined by whether a correct CLOM result is generated using the translation (given such a mapping exists in the given CLOM scenario).
- **AOLT process** and **AOLT selection process** are used interchangeably in this thesis. Both refer to one realisation (among others) of the AOLT concept.
- **AOLT component** refers to a system component that is an integrated AOLT process within the CLOM system such as SOCOM or SOCOM++.
- **Ontological resources** and **entities** are used interchangeably in this thesis. Both refer to any formally defined conceptualisation that is identifiable with a unique

resource identifier (URI) in the give ontology. Ontological resources include classes, properties and individuals.

- **Ontology label translation** refers to the translation of the natural language segment used to identify an ontological resource. For example, *CommunityStatus* in <owl:Class rdf:about= "http://swrc.ontoware.org/coin# CommunityStatus"/> would be translated in order to apply MOM techniques in the process of achieving CLOM. Note that the ontology label translation process does not translate the natural language content of RDFS vocabularies[8]. For example, *List* from <rdfs:Class rdf:about ="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"> would not be translated since it is a syntax specification.

1.8. Thesis Overview

A DVD (digital versatile disc) is submitted along with this thesis, which contains the Java code used for the implementations of the baseline system, SOCOM and SOCOM++. Raw data collected from all the experiments shown in this thesis can also be found on this disk. A table of content for this DVD can be found in appendix A. This thesis contains East Asian and European characters, additional support packs[9] may be required to display these languages correctly.

The remainder of this thesis is organised as follows. Chapter 2 discusses the state of the art in cross-lingual ontology mapping, and presents some background knowledge on monolingual ontology mapping, machine translation and mapping evaluation metrics.

Chapter 3 investigates a translation-based approach to cross-lingual ontology mapping that was identified in chapter 2. An implementation of this approach: the baseline system, is evaluated through a set of CLOM experiments (involving ontologies in Chinese and English).

Motivated by the conclusions drawn from the experimental findings in chapter 3, chapter 4 proposes the AOLT concept, and the design, implementation, evaluation of the SOCOM system that implements a basic process to realise the AOLT concept. The evaluation carried out on SOCOM aims to validate the AOLT concept through two

[8] For a list of RDFS vocabularies, see http://www.w3.org/TR/rdf-schema/rdfs-namespace

[9] For Microsoft Office Word service packs, please see http://office.microsoft.com/en-us/word-help/ CH006083250.aspx?CTT=97

CLOM experiments (involving ontologies in Chinese, English and French). In addition, the SOCOM system is applied in a cross-lingual personalised document retrieval system (involving ontologies in English and German) to showcase the potential application use of the SOCOM system.

Motivated by the positive findings from the initial prototype, chapter 5 presents a second prototype: SOCOM++. SOCOM++ integrates a flexible AOLT process, with the purpose of adjusting its configurable inputs to influence the CLOM outcome. Six trials (involving ontologies in English, Chinese and French) and two scalability tests (involving ontologies in English and Japanese) of SOCOM++ (with different AOLT configurations) were carried out to demonstrate how appropriate translation selections can be adjusted for the same ontology pair in a CLOM scenario, and how execution time is affected given increased workload. The flexibility of the AOLT component is demonstrated through these trials, and the evaluation results on the mappings generated showed a range of quality achieved in experiments.

Finally, chapter 6 concludes this thesis with a summary of the research objectives achieved and contributions of this research, and suggests several future research directions in the area of cross-lingual ontology mapping.

2 BACKGROUND AND STATE OF THE ART

2.1. Chapter overview

Related background and a state of the art review are presented in this chapter. The ontology mapping problem is first introduced and defined in section 2.2. As mentioned previously in chapter 1 (section 1.2), ontologies are not always authored in the same natural language, multilinguality in ontologies is thus discussed in section 2.3. Enabling semantic interoperability among multilingual ontologies is a major driver for the development of cross-lingual ontology mapping (CLOM). Section 2.4 presents a survey of the current approaches to CLOM. A popular approach to CLOM, namely translation-based CLOM, is identified in this survey that integrates monolingual ontology matching (MOM) techniques and machine translation (MT) techniques. A brief overview on MOM techniques is thus followed in section 2.5, and a brief background on MT techniques is presented in section 2.6. Common evaluation methods currently used in ontology mapping research are discussed in section 2.7. Finally, section 2.8 concludes this chapter with a summary.

2.2. The Ontology Mapping Problem

The promise of the semantic web is that of a new way to organise, present and search information that is based on the meaning (i.e. semantics that can be manipulated by machines) and not just text (i.e. unstructured information designed for humans to process) [Berners-Lee et al., 2001]. To model meaning in a structured fashion, ontologies have gained increasing interest from the semantic web community [Maedche

& Staab, 2001]. However, in decentralised environments such as the semantic web, the heterogeneity issue occurs when ontologies are authored by different actors. This issue can be thought of in a similar manner to the database management problem, where database administrators use different terms to store the same information in different database systems. As mentioned in chapter 1 (section 1.3), ontologies are specifications of conceptualisations [Gruber, 1993], which implies that ontologies are subjectively constructed. This means that views on the same domains of interest will differ from one person to the next, depending on their conceptual model and background knowledge for example. To address the heterogeneity issue arising from ontologies on the semantic web, ontology mapping has become an important research field [De Bruijn et al., 2007].

In the literature, *ontology matching* (e.g. [Euzenat & Shvaiko, 2008]), *ontology mapping* (e.g. [Kalfoglou & Schorlemmer, 2003]) and *ontology alignment* (e.g. [Ehrig, 2007]) are used interchangeably to refer to the process of correspondence generation between ontologies. The concept of ontology matching and the concept of ontology mapping are differentiated in [O'Sullivan et al., 2007], whereby the former refers to the identification of candidate matches between ontologies and the latter refers to the establishment of actual correspondences between ontological resources based on candidate matches. Following the approach proposed by O'Sullivan et al., in this thesis, ontology mapping is viewed as a two-step process, whereby the first step involves the generation of candidate correspondences (i.e. pre-evaluation) and the second step involves the generation of validated correspondences (i.e. post-evaluation). The outcome from step one is referred to as *candidate matches*, and the outcome from step two is referred to as *mappings* in this thesis. The implemented prototypes: SOCOM and SOCOM++ presented in this thesis aim to provide support to the cross-lingual ontology mapping process by generating candidate matches through the matching process.

> The following definition for ontology matching is adopted by this thesis:
> "*The matching process can be seen as a function f which, from a pair of ontologies to match o and o', an input alignment A, a set of parameters p and a set of oracles and resources r, returns an alignment A' between these ontologies: A' = f (o, o', A, p, r)*" [Euzenat & Shvaiko, 2007 p.44]

The goal of the mapping process is to generate correspondences between ontology resources, whereby the following definition for correspondence is adopted in this thesis:

"*Given two ontologies o and o' with associated entity languages O_L and $Q_{L'}$, a set of alignment relations Θ and a confidence structure over Ξ, a correspondence is a 5-uple:* ⟨ *id, e, e', r, n* ⟩ *, such that id is a unique identifier of the given correspondence;* $e \in O_L(o)$ *and* $e' \in Q_{L'}(o')$*;* $r \in \Theta$*;* $n \in \Xi$*. The correspondence* ⟨ *id, e, e', r, n* ⟩ *asserts that the relation r holds between the ontology entities e and e' with confidence n.*" [Euzenat & Shvaiko, 2007 p.46][10]

A set of alignment relations "*correspond to set-theoretic relations between classes: equivalence (=); disjointness (⊥); more general (⊒) … relations can be of any type and are not restricted to relations present within the ontology language, such as fuzzy relations or probability distributions over a complete set of relations or similarity measures*" [Euzenat & Shvaiko, 2007 p.45]. A confidence structure is "*an ordered set of degrees* ⟨ *Ξ, ≤* ⟩ *for which there exists a greatest element* T *and a smallest element* ⊥" [Euzenat & Shvaiko, 2007 p.46]. In this thesis, MOM results are generated using the Alignment API (discussed in section 2.5.2) and CLOM results are generated based on these MOM results (more on this in section 2.5). In the experiments shown in this thesis, the Alignment API only generates equivalence relations, where correspondences are equivalent images of one another with confidence levels that range between 0.0 and 1.0. Equivalent correspondences are currently the dominate relations that are generated by MOM tools - thus is the focus of this research - this is evidently shown by the participating MOM systems in the ontology alignment evaluation initiative (OAEI) contests since 2004[11].

Ontologies are likely to be authored by different actors who not only have differing conceptualisations of the world but also different natural language preferences. Multilinguality is an inevitable characteristic of ontologies. A brief overview on recent research related to multilingual ontologies is discussed next.

2.3. Ontologies and Multilinguality

Ontologies are widely used in knowledge-based systems and the applications of ontologies traverse many disciplines. Five example use of ontologies in the field of

[10] In this context, entity language refers to the ontology language, e.g. OWL, RDF, etc. In this thesis, natural language refers to linguistic languages possessed by humans.
[11] OAEI results since 2004 can be found at http://oaei.ontologymatching.org

agriculture, education, e-learning, finance and medicine are discussed here. In agriculture, the Food and Agriculture Organization [12] (FAO) provides reference standards for defining and structuring agricultural terminologies. Since all FAO official documents must be made available in five official languages including Arabic, Chinese, English, French and Spanish, a large amount of research has been carried out on the translations of large multilingual agricultural thesauri [Chang & Lu, 2002], mapping methodologies for them [Liang et al., 2005; Liang & Sini, 2006] and a definition of requirements to improve the interoperability of these multilingual information resources [Caracciolo et al., 2007]. In education, the Bologna declaration has introduced an ontology-based framework for qualification recognition [Vas, 2007] across the European Union (EU). In an effort to best match labour markets with employment opportunities, an ontology is used to support the recognition of degrees and qualifications within the EU (which consists of 27 member states and 23 official languages in 2011[13]). In e-learning, educational ontologies are used to enhance learning experiences [Cui et al., 2004] and to empower system platforms with high adaptivity [Sosnovsky & Gavrilova, 2006]. In finance, ontologies are used to model knowledge in the stock market domain [Alonso et al., 2005] and portfolio management [Zhang et al., 2000]. In medicine, ontologies are used to improve knowledge sharing and reuse, such as work presented by Fang et al. [Fang et al., 2006] which focuses on the creation of a traditional Chinese medicine ontology, and work presented by Tenenbaum et al. [Tenenbaum et al., 2011] which focuses on the development of the Biomedical Resource Ontology in biomedicine. A key observation from ontology-based applications such as those mentioned above is that the development of ontologies is closely associated with natural languages. Given the diversity of natural languages and the different conceptual models of ontology engineers, the heterogeneity issue is inevitable in the presence of ontologies that are built on different models of conceptualisations and varied natural languages. The very existence of ontologies in various natural languages provides an impetus to discover ways to support the necessary semantic interoperability for the purpose of knowledge sharing.

[12] http://www.fao.org

[13] In 2011, EU member states include Austria, Belgium, Bulgaria, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden and United Kingdom. The official working natural languages of the EU include Bulgarian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Irish, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovene, Spanish and Swedish.

Thesauri - often containing structured terms that are synonyms and antonyms of one another - can be considered as light weight ontologies. According to the Global WordNet Association[14], at the time of this writing, there are more than forty thesauri in the world containing a collective set of over fifty different natural languages. These languages include Arabic (used in ArabicWordNet[15]); Bulgarian (used in BulNet[16]); Chinese (used in HowNet[17]); Dutch, French, German, Italian, Spanish (used in EuroWordNet[18]); Irish (used in LSG[19]) and many others. Multilinguality is also evident in formally defined ontologies. According to the OntoSelect Ontology Library[20], (at the time of this writing) more than 25% of indexed 1530 ontologies are written in natural languages other than English. With the rise of multilinguality in ontologies, research effort dedicated to supporting the generation of multilingual ontologies can be seen. For example, Lauser et al. [Lauser et al., 2002] introduce a semi-automatic framework in an attempt to reduce labour costs. Niwa et al. [Niwa et al., 1997] define a formula to extract word relations based on document frequency and conditional probability. Srinivasan [Srinivasan, 1992] conducted similar research and proposed an algorithm to generate hierarchies of words. Shimoji & Wada [Shimoji & Wada, 2008] propose a method that creates a hierarchy of words based on natural language contents from an English-Japanese dictionary, and shows that their method renders more refined hierarchy relationships than the previous two methods. These notable research projects highlight various support that is available for the creation of multilingual ontologies. However, not a lot of attention has been devoted to supporting the interoperability of multilingual ontologies. Research efforts to date that aim to tackle the cross-lingual ontology mapping issue are discussed next.

2.4. Cross-Lingual Ontology Mapping

This section presents the state of the art in CLOM. Five categories of CLOM approaches are discussed in section 2.4.1. A popular approach to CLOM, namely the translation-based approach to CLOM is identified through this review. An important note regarding the translation-based approach is the distinction between translations

[14] http://www.globalwordnet.org
[15] http://www.globalwordnet.org/AWN
[16] http://dcl.bas.bg/BulNet/general_en.html
[17] http://www.keenage.com
[18] http://www.illc.uva.nl/EuroWordNet
[19] http://borel.slu.edu/lsg
[20] http://olp.dfki.de/ontoselect

that are carried out for the purpose of CLOM and translations that are carried out for the purpose of ontology localisation. This is discussed in section 2.4.2.

2.4.1. Categories of CLOM Approaches

Current approaches to CLOM can be grouped into five categories, namely: manual CLOM [Liang & Sini, 2006], corpus-based CLOM [Ngai et al., 2002], CLOM via linguistic enrichment [Pazienta & Stellato, 2005], CLOM via indirect alignment [Jung et al., 2009] and translation-based CLOM [Wang et al., 2009; Trojahn, 2008; Zhang et al., 2008]. Each category is discussed next.

Manual CLOM refers to those approaches that rely solely on human experts whereby mappings are generated by hand. An example of manual CLOM is discussed in [Liang & Sini, 2006], where an English thesaurus: AGROVOC[21] (developed by the FAO containing a set of agricultural vocabularies) is mapped to a Chinese thesaurus: CAT [22] (Chinese Agricultural Ontology, developed by the Chinese Academy of Agricultural Science) by hand. The thesauri are loaded in the Protégé editor, and segments of the thesauri are assigned to groups of terminologists to generate mappings. Finally, these manually generated mappings are reviewed and stored. Liang & Sini did not propose an evaluation method for their work. However, it can be understood that since mappings are generated by human experts and are reviewed, that they are effectively evaluated and are of good quality. The advantage of this approach is that the mappings generated are likely to be accurate and reliable. However, given large and complex ontologies, this can be a time-consuming and labour-intensive process.

Corpus-based CLOM refers to those approaches that require the assistance of bilingual corpora when generating mappings. Such an example is presented in [Ngai et al., 2002]. Ngai et al. use a bilingual corpus to align WordNet (in English) and HowNet (in Chinese). The bilingual corpus is created using newspaper content (in English and Chinese) and term frequency analysis (i.e. vector-based co-occurrence studies of words that appear together in the corpus) are carried out to associate synsets[23] in the given thesauri. Finally, the evaluation of their approach is conducted by a team of two domain

[21] http://aims.fao.org/website/AGROVOC-Thesaurus/sub
[22] http://www.ciard.net/partners/labof-chinese-agricultural-ontology-services
[23] A synset is a synonym set, which can be defined as "*a set of words that are interchangeable in some context without changing the truth value of the proposition in which they are embedded*" ~ WordNet Reference Manual, Princeton University, at http://wordnet.princeton.edu/wordnet/documentation/

experts. The advantage of this approach is that the corpora need not be parallel (unlike corpus-based statistical MT whereby parallel corpora are often required [Koehn, 2005]), which makes the construction process easier. However, a disadvantage of using corpora is that the construction overhead could be a costly process for domain-specific ontologies. In addition, Ngai et al.'s approach heavily relies on synsets, which is a requirement that can often be satisfied by thesauri, but not necessarily by formally defined ontologies in OWL or RDF.

CLOM via linguistic enrichment: Pazienza & Stellato [Pazienta & Stellato, 2005] propose a linguistically motivated mapping approach and urge linguistically motivated ontology development, whereby ontologies would contain human-readable linguistic resources that can offer strong evidence in the mapping process. To facilitate this process, the OntoLing plug-in [Pazienza & Stellato, 2006b] was developed for the Protégé editor. The plug-in presents an interface to the ontology engineer during the ontology development, whereby word senses (e.g. extracted from WordNet) can be associated to ontological resources. Lastly, precision, recall and f-measure (these measurements are discussed in detail in section 2.7) are used to measure Pazienta & Stellato's system. Linguistic enrichment of ontological resources will offer strong evidence in the process of mapping generation. However, as already pointed out by the authors, this enrichment process is currently unstandardised. As a result, it can be difficult to build CLOM algorithms based upon these linguistically enriched ontologies.

CLOM via indirect alignment can be classified as a form of mapping reuse. This is a concept that already exists in MOM as alignment reuse and repository of structures (see section 2.5, figure 2-1). In the context of CLOM, indirect alignment refers to the process of generating new CLOM results using pre-existing CLOM results. Such an example is given in [Jung et al., 2009]. Jung et al. present indirect alignment among ontologies in English, Korean and Swedish, given alignment A which is generated between ontology O_1 (e.g. in Korean) and O_2 (e.g. in English), and alignment A' which is generated between ontology O_2 and O_3 (e.g. in Swedish). Then mappings between O_1 and O_3 can be generated by reusing alignment A and A' since they both concern one common ontology O_2. An evaluation of Jung et al.'s proposal is presented in [Jung, 2011] whereby precision and recall are used to measure mapping quality. Assuming the availability of A and A', this is an easy approach to achieve technically. However, as this technique requires the very existence of A and A' which currently remains a challenge in itself, it can be difficult to apply this approach in some CLOM settings.

Translation-based CLOM refers to the use of translation techniques (which can be achieved through the use of MT tools, bilingual/multilingual thesauri, dictionaries etc.) in the CLOM process. Typically in translation-based CLOM approaches, a CLOM problem is converted to a MOM problem first, which is then solved using MOM techniques next. Compared to previously discussed approaches, the translation-based CLOM is currently a very popular approach that is exercised by several researchers (see table 2-1), mostly due to its simplicity to execute and the vast number of readily available tools in MT and MOM. The translation-based CLOM approach is already shown to be feasible in the state of the art, however, the impact of translations on the final mapping outcome has not yet been investigated. This thesis aims to fill this research gap and provide better support for MT and MOM tools in the process of CLOM. In this thesis, the translation-based CLOM approach is referred to as the baseline approach, which serves as a basis in the evaluation of the proposed solution (discussed in chapter 4, section 4.2). Five examples of translation-based approach to CLOM are discussed next, including three test cases from the OAEI contests and two others from outside the OAEI community. Table 2-1 presents a summary of these translation-based CLOM approaches in the state of the art.

Table 2-1. An Overview of Translation-Based CLOM Approaches

Approach	Translation Means	Matching Means
Zhang et al.	Bilingual dictionary	The RiMOM tool
Bouma	Multilingual thesaurus & bilingual encyclopedia	The GG2WW tool
Nagy et al.	DBpedia	The DSSim tool
Wang et al.	GoogleTranslate online service	Instance-based matching tool
Trojahn et al.	GoogleTranslate API	The Alignment API

The OAEI introduced its first ontology mapping test case involving different natural languages in 2008. The OAEI mldirectory test case[24] consists of matching web site directories (including Dmoz, Licos and Yahoo) in different languages (i.e. English and Japanese). Zhang et al. [Zhang et al., 2008] used a Japanese-English dictionary to first translate the labels in the Japanese web directory into English. They then carried out monolingual matching procedures using the RiMOM[25] tool. It should be noted that among 13 participants in 2008, only one contestant (i.e. RiMOM) submitted results from this test case. These results however were not evaluated by the OAEI[26]. The outcome from the mldirectory test case shows a lack of attention on CLOM from the

[24] The data set is available at http://oaei.ontologymatching.org/2008/mldirectory
[25] RiMOM's homepage can be found at http://keg.cs.tsinghua.edu.cn/project/RiMOM/
[26] A record of the number of matches generated was published at http://oaei.ontologymatching.org/2008/results/mldirectory/. However, evaluations on these matches were never conducted.

ontology mapping community, and highlights the urgency of supporting ontology mappings that are carried out in the multilingual environment.

OAEI 2009 introduced a VLCR (Very Large Cross-lingual Resources) track involving the mappings of thesauri in Dutch (GTAA – Thesaurus of the Netherlands Institute for Sound and Vision) and English (WordNet and DBpedia)[27]. Among 16 participants, only 2 contestants submitted results. Bouma [Bouma, 2009] uses the multilingual EuroWordNet (which includes synsets in English and Dutch) and the Dutch Wikipedia to bridge between Dutch and English. Mappings between the GTAA thesaurus to WordNet and DBpedia are then generated using the GG2WW tool in the monolingual environment. Nagy et al. [Nagy et al., 2009] uses DBpedia itself to associate concepts in English and Dutch, since the articles and titles in DBpedia are often labelled in both natural languages. Mappings are finally generated using the DSSim tool in the monolingual environment. Partial evaluations on the matches generated from these two systems were conducted by the OAEI. More specifically, random sample matches (some 71-97 matches are randomly selected from 3663 matches generated by GG2WW, and from 2405 matches generated by DSSim) are evaluated based on a partial gold standard (including 100 reference mappings) using precision and recall[28]. A greater recall was found in the GG2WW tool (around 0.6) comparing to the DSSim tool (around 0.2). However, precision of both systems varied greatly. The GG2WW system neglected specific matches such as mappings between GTAA locations to WordNet locations (leading to a range of precision scores between 0.0 and 0.9). Though the DSSim tool did not neglect any specific types of match, however its precision scores ranged largely (between 0.1 to 0.8). Although the evaluation was only partially conducted, it nevertheless offers some insight into the quality of these matches. One key conclusion from this test case is that the quality of the matches is noticeably poorer than those generated in the monolingual environment. For example, in the benchmark data set of the same year (where mappings are carried out between English ontologies), the DSSim tool was able to generate matches yielding a much higher average precision (0.97) and recall (0.66). It is not known whether this was seen with the GG2WW tool, as it only took part in the VLCR test case.

The VLCR test case was again included in the OAEI 2010 contest, where only one tool (RiMOM) took part among a total of 16 contestants. Wang et al. present a

[27] The VLCR test case can be found at http://oaei.ontologymatching.org/2009/vlcr/
[28] The evaluation results can be found at http://oaei.ontologymatching.org/2009/results/vlcr/

record of the number of matches generated by RiMOM in [Wang et al., 2010] and describe an instance-based matching approach at a very high level (it is not clear whether the same translation technique presented in OAEI 2008 was used for this test case). However, these matches were never evaluated by the authors or by the OAEI. Although the VLCR homepage states matching samples are to be evaluated in the same fashion as in the previous year, the evaluation results have not been published[29]. At the time of this writing, OAEI 2011 is in its preparation stage. However, it is not known at present time, whether the final test cases will include multilingual data sets[30].

There has been some effort outside the OAEI community that tackles the CLOM problem by applying translation techniques. In particular, work of Wang et al. [Wang et al., 2009] and Trojahn et al. [Trojahn, 2010] are discussed next. Wang et al. [Wang et al., 2009] use the GoogleTranslate service to translate digital library vocabularies before applying instance-based matching techniques to generate mappings among library subjects written in English, French and German. To evaluate the matches, a manually generated gold standard was used. However, only precision scores were calculated in the evaluation due to the incomplete gold standard (as it was still being created at the time). The partial evaluation showed the precision ranged between 0.4 and 0.8. However, the recall of these results is unknown (without a complete gold standard). Wang et al.'s work presents a similar strategy to CLOM as those deployed in RiMOM, DSSim and GG2WW, whereby machine translation technique is applied instead of dictionaries or thesauri. For all of them, the goal is to convert a cross-lingual mapping issue into a monolingual mapping issue, which can then be solved with MOM techniques. A similar approach is presented by Trojahn et al. [Trojahn, 2010], which incorporates the work presented in [Fu et al., 2009; Jung et al., 2009]. CLOM is achieved by first applying the GoogleTranslate API to bridge between different natural languages which is then followed by MOM techniques. In addition, their tool is accompanied by a mapping reuse feature as presented in [Jung et al., 2009]. Trojahn et al.'s approach is evaluated using ontologies in English, French and Portuguese through the generation of precision, recall and f-measure scores. A range of precision (0.41-0.86), recall (0.05-0.51) and f-measure scores (0.10-0.62) were achieved.

---

[29] Detail data set description and evaluation strategies of the VLCR test case in 2010 can be found at http://oaei.ontologymatching.org/2010/vlcr/index.html

[30] An overview of OAEI 2011 test cases can be found at http://oaei.ontologymatching.org/2011/. At the time of this writing (July 2011), all published data sets (a total of seven tracks) involve just the English language.

A key common characteristic shared by translation-based CLOM approaches discussed above is that CLOM is achieved through two steps. Translations are first carried out to bridge between the natural languages in the given ontologies. This is then followed by MOM techniques next. What is evident from the state of the art is that existing research in CLOM has successfully demonstrated the feasibility of incorporating MT and MOM techniques. However, little effort is made to investigate whether there is a positive/negative impact from the translation process on the subsequent MOM process. So far, these research studies have shown that MT and MOM techniques can be applied in the CLOM process, however, it is not clear to what extent these techniques can be incorporated to support the generation of high quality CLOM results. This thesis aims to investigate this un-tackled issue. An important point to note is that the translations taken place in the context of CLOM differs from the translations taken place in the context of ontology localisation. This is discussed next.

2.4.2. Translations in CLOM vs. Translations in Ontology Localisation

Translations of natural language content presented in ontologies are studied in the field of ontology localisation[31]. Ontology localisation is defined as "the adaptation of an ontology to a particular language and culture" [Suárez-Figueroa & Gómez-Pérez, 2008]. This definition is further refined by Cimiano et al. as "the process of adapting a given ontology to the needs of a certain community, which can be characterised by a common language, a common culture or a certain geo-political environment" [Cimiano et al., 2010]. Cimiano et al. point out that the ontology localisation process takes place at the lexical layer, the conceptualisation layer as well as the interaction between these layers (i.e. the changes in one layer may influence the changes in the other layer). In other words, the ontology localisation process goes beyond than simply localising the labels (i.e. at the lexical layer), but the structure of the ontologies may also be changed in order to adapt to the target community and its culture (i.e. at the conceptualisation layer). Note that translation is a step towards localisation but is not equal to localisation,

[31] Note that ontology localisation differs from ontology translation. Ontology translation refers to "*the translation of a dataset from one ontology to another… The translation problem arises when web-based agents try to exchange their datasets but they use different ontologies to describe them*" [Dou et al., 2004], e.g. translating an ontology formatted in DAML (DARPA Agent Markup Language) to OWL (Web Ontology Language). More details of ontology translation can be found in [Chalupsky, 2000].

since translation removes the natural language barrier but not necessarily the culture barrier.

To facilitate the localisation of ontology labels (i.e. the identifiers of ontological resources that are in the natural language format), tools such as the LabelTranslator tool [Espinoza et al., 2008] has been developed. The LabelTranslator tool provides suggested candidate translations for labels of a given ontology (which are selected one at a time by the user) in one of three natural languages, English, Spanish and German. The goal of the LabelTranslator tool is to aid the user to better understand the semantics presented in the given ontology, as it presents a form of description of the ontological resources in the natural language that is preferred by the user.

The work presented in this thesis, though it involves translations of ontology labels, however is different from the work presented by Espinoza et al.. First of all, the motivation for the LabelTranslator tool is ontology localisation, whereas the motivation for this thesis is cross-lingual ontology mapping whereby translations are merely an intermediate step to the actual goal of generating mappings between the given ontologies. The localisation of ontologies may involve rearranging the structures of the ontological resources [Cimiano et al., 2010] as well as editing the labels (although structural changes are not yet supported by the LabelTranslator tool at the time of this writing). It is useful to note that the approach presented in this research does not attempt changing the existing structures of ontologies. Secondly, given the different motivations, the perceived goals of this research and the LabelTranslator tool differ significantly. The LabelTranslator tool aims to suggest translations that adapt to the target communities, whereby the final translations are selected for the purpose of localisation. In contrast, this work aims to improve the quality of CLOM whereby the final translations are selected in a way to enable the generation of high quality CLOM results. Lastly, the LabelTranslator tool requires the involvement of users in the selection of the final localised labels; whereas the selection of translations is automated in the work presented in this thesis.

As mentioned in chapter 1 (section 1.7), ontology labels in this thesis refer to the identifying labels of an ontological resource. In other words, these are strings that are used to name ontological resources in a formally defined ontology. For example, in `<Class rdf:ID="Thing"/>`, *Thing* is the ontology label of this defined class. Another example can be `<owl:Class rdf:about="http://swrc.ontoware.org/ontology#`

`Person"/>`, where `http://swrc.ontoware.org/ontology#` is the namespace declared for the class that has a label *Person*. It should not be confused with `rdfs:label`, whereby declarations such as `<rdfs:label>Thing</rdfs:label>` are often used to associate one named resource with additional labels. `rdfs:label` can be used to tag multiple (multilingual) natural language labels to a particular resource (as the example shown in chapter 1, figure 1-1), however, they cannot be used to identify a named resource. In this thesis, ontology label translation refers to the process of translating natural language content that is used to identify ontological resources.

In summary, a key observation from the review on the state of the art in CLOM is that, it is evident from the OAEI contests and other related research efforts discussed above that the field of CLOM has not received much attention from the ontology mapping community. Although three tools have participated in multilingual test cases in OAEI to date, it is difficult to evaluate the success of these tools when they participate in different test cases and especially when their results have not been evaluated thoroughly. Using MT as a means to bridge the gap between natural languages is a feasible approach to achieve CLOM as shown in the literature. However, it is not yet a thoroughly examined method. How good are the translations returned from MT tools? Are these translations suitable for the MOM tools in the process of achieving CLOM? How will these translations impact on the final quality of the mappings? Can CLOM quality be improved given appropriate translations? These questions are currently unanswered in the state of the art, and with this thesis the aim is to contribute towards the answering of them. Next, some background on the related fields - namely MOM and MT - to achieve translation-based CLOM is discussed.

2.5. Monolingual Ontology Mapping

This section presents a brief background overview on MOM. Section 2.5.1 presents two ways to categorise current MOM techniques. Section 2.5.2 discusses the MOM tool, namely the Alignment API, that is used in this thesis.

2.5.1. Categories of MOM Techniques

Ontology mapping in the monolingual environment is a well-studied research field, where various matching tools (a survey of MOM tools is presented in [Euzenat &

Shvaiko, 2007a]) and interfaces (e.g. Optima [Kolli & Doshi, 2008], CogZ [Falconer & Storey, 2007]) have been developed to facilitate the mapping process. Since 2004, contests organised by the Ontology Alignment Evaluation Initiative[32] (OAEI) have been held on an annual basis. The OAEI contests provide datasets and gold standards in an effort to evaluate and improve participating mapping systems. Some of these datasets are used for the evaluation of the research work presented in this thesis (discussed in chapter 4 and 5). Several surveys of current MOM tools and classifications of MOM techniques are available in the literature such as [Euzenat & Shvaiko, 2007; Shvaiko & Euzenat, 2008; Giunchiglia et al., 2007; Shvaiko & Euzenat, 2005; Kalfoglou & Schorlemmer, 2003]. This section aims to provide a brief overview.

Euzenat & Shvaiko present an extensive review on MOM techniques and systems in [Euzenat & Shvaiko, 2007]. One way to categorise MOM techniques is based on how input is interpreted and its granularity, whereby MOM techniques can be grouped into two broad categories: element-level and structure-level, as shown in figure 2-1. Each category is discussed next.

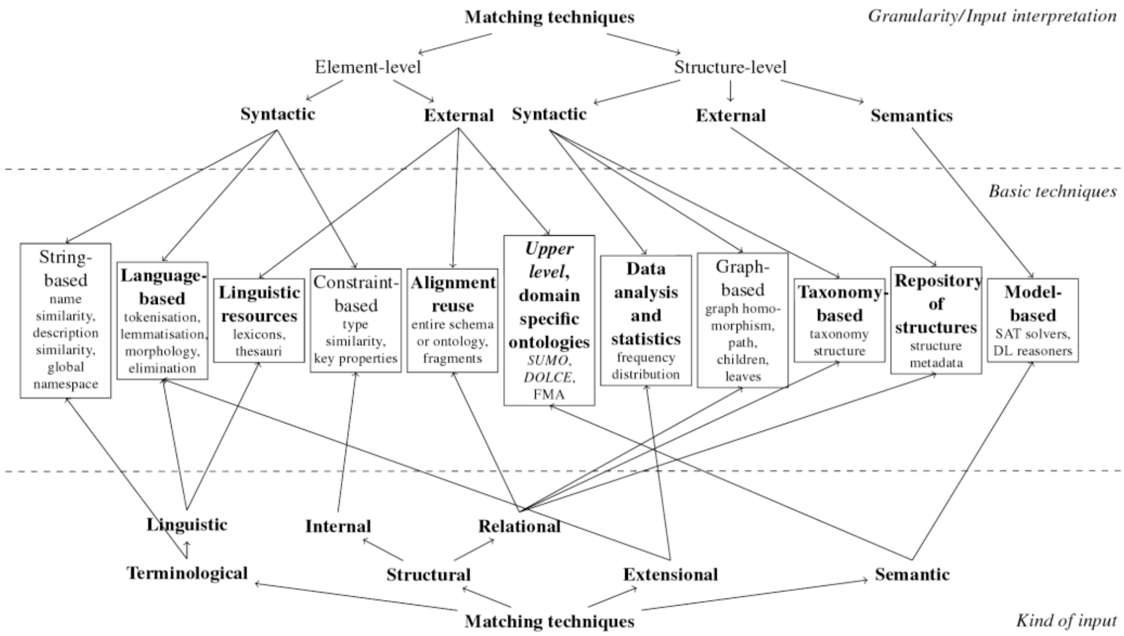


Figure 2-1. Euzenat & Shvaiko's Classification of Matching Approaches [Euzenat & Shvaiko, 2007 p.65]

At the element level, matches are computed by “analysing entities or instances of those entities in isolation, ignoring their relations with other entities or their instances” [Euzenat & Shvaiko, 2007 p.64]. Examples of element-level matching techniques include those that are string-based, language-based or constraint-based and those that

[32] http://oaei.ontologymatching.org

apply linguistic resources and reuse existing mappings. String-based techniques apply methods such as edit distance string comparison to conclude the similarity between two strings, whereby the smaller the number of edits required to turn one string into the other, the more similar these strings are to one another. Language-based techniques apply methods such as natural language processing procedures (e.g. extracting meanings of words from dictionaries) to conclude string similarities, whereby strings are treated as units of texts rather than sequences of character (as used in string-based techniques). Constraint-based techniques take internally defined restrictions on ontological resources (e.g. cardinality, range defined for a property with respect to a particular ontological class) in the process of concluding correspondences. Techniques that use linguistic resources often apply thesauri and lexicons (e.g. WordNet) in the process of establishing correspondences. Finally, matches can be generated based on previously concluded mappings (that are either partial fragments of a complete match set or entire set). Whether correspondences are concluded based on existing mappings, or based on comparisons made between sequences of characters or sequences of words, or from comparing constraints declared or linguistic evidence that is available for the given resources, the aforementioned techniques have one key attribute in common - is that they assume comparisons take place in the context of comparable natural languages. For example, string comparisons made between *Conference* and *ConferenceVenue* (both in English), or *Conference* and *Konferenz* (meaning "conference" in German) are likely to conclude that the two terms in the given pair are somewhat similar to each other. This is because these terms are in natural languages derived from the same language family or at least use the same graphemes (see footnote 4 in chapter 1). In this case, both English and German belong to the Germanic language family. Such comparison techniques however, do not apply to natural languages that do not share the same graphemes. For example, string-based techniques cannot compute similarity measures between *Cheese* and *奶酪* (meaning "cheese" in Chinese), even though they contain the same meaning. This limitation of MOM techniques clearly needs to be addressed in the context of cross-lingual ontology mapping.

In contrast to element-level techniques, at the structural level, matches are computed by "analysing how entities or their instances appear together in a structure" [Euzenat & Shvaiko, 2007 p.64]. Examples of structure-level techniques include those that are graph-based, taxonomy-based, model-based or statistic-based and those that use repository of structures. Graph-based techniques analyse the positions of nodes in a given ontological structure (which is considered as a graph) in the process of concluding correspondences, whereby methods such as maximum common directed subgraph [Bunke & Kandel, 2000] are applied. For example, to compare the semantics of two classes C_1 (from ontology O_1) and C_2 (from ontology O_2), the sub-classes of C_1 are compared to the sub-classes of C_2. Taxonomy-based techniques also apply graph algorithms but only consider the *is-a* relations in the given ontologies in the process of concluding correspondences. The main reasoning behind taxonomy-based techniques is that if *is-a* relations already associate two resources, then the surrounding nodes of these resources should also be similar. Model-based techniques generate correspondences based on comparisons made on the semantic interpretations, which often require background knowledge such as topic ontologies with comprehensive coverage of the domains of interest. Such an example of external topic ontologies is the Suggested Upper Merged Ontology (SUMO) in [Niles & Pease, 2001]. Statistic-based techniques apply statistical methods to generalise regularities and discrepancies when concluding correspondences. For example, given class C_1 (from ontology O_1) and class C_2 (from ontology O_2), assuming they both contain a set of instances of their own, if statistical analysis suggests a large number of instances from the two sets are similar, then it is likely that their corresponding classes C_1 and C_2 are also similar. Finally, techniques that use repositories of structures make use of repositories that contain similarities between ontologies (not similarities between resources as in mapping reuse) in order to conclude correspondences. An example of such is presented in [Rahm et al., 2004], where previously concluded similar fragments are used to denote similarities between new structures. A common characteristic among matching techniques that take structures into account during the mapping process is that, in order to compare sets of sub-classes, relations, or structure fragments, their associated labels (i.e. identifiers of these sub-classes, relations and structure fragments) – often in natural language form – need to be compared. This means that structure-level techniques often will require the assistance from element-level techniques. As discussed earlier, element-level techniques are commonly limited to mapping environments involving comparable natural languages. This implies that given multilingual mapping environment, structure-level techniques will encounter difficulty considering they are effectively built upon conclusions from element-level techniques. In fact, this trend is shown through the evaluations presented in chapter 3.

Ehrig [Ehrig, 2007] presents another classification of MOM techniques, and concludes three layers of similarity in ontology mapping: the data layer, the ontology layer and the context layer, with an orthogonal dimension that represents specific domain knowledge at all layers, as shown in figure 2-2. At the data layer, comparisons are made by “considering data values of simple or complex datatypes such as integers and strings”. Techniques used at the data layer include edit distance string comparison and relative distance comparison (i.e. distance relative to a specified reference point). Matching strategies at this layer are similar to the aforementioned string-based techniques which are classified under element level by Euzenat & Shvaiko. At the ontology layer, “semantic relations between the entities” are compared, which range from the graph level (similar to the aforementioned graph-based techniques classified under structure-level in [Euzenat & Shvaiko, 2007 p.69]), to the description logics level (similar to the aforementioned taxonomy-based techniques classified under structure-level in [Euzenat & Shvaiko, 2007 p.69]), then to the restriction level (similar to the aforementioned constraint-based techniques classified under element-level in [Euzenat & Shvaiko, 2007 p.67]) and finally to the rule level (where high level reasoning is conducted upon existing rules [Fürst & Trichet, 2005]). At the context layer, comparisons are made between resources based on their usages in external applications (this expands on Euzenat & Shvaiko’s classifications of techniques that uses linguistic resources and are language-based). Finally, the orthogonal dimension illustrates domain knowledge which can be inserted into any of the three layers. Ehrig’s view on domain knowledge is similar to the Euzenat & Shvaiko’s view on matching strategies that use external resources such as existing mappings (categorised as alignment reuse under element level techniques) and repositories of structures (categorised as repository of structures under structure level techniques).

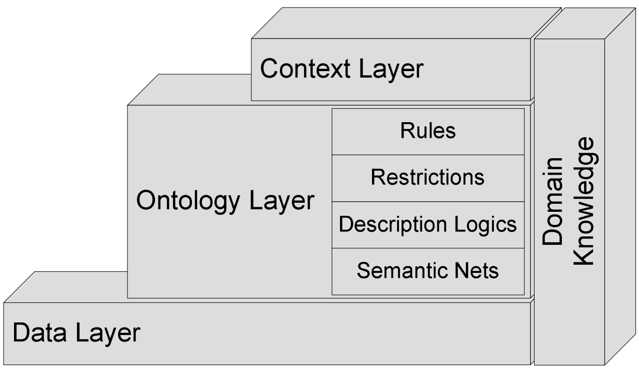


Figure 2-2. Ehrig's Similarity Layers [Ehrig, 2007 p.27]

A key observation emerging from the work presented by Euzenat & Shvaiko and Ehrig is that there is a rich set of MOM techniques that are currently available. This diversity of techniques highlight the extensive research in the field of MOM to date. This is also reflected in the development of MOM tools and systems. Choi et al. [Choi et al., 2006] surveyed nine MOM tools and compared them to one another based on input, output, interaction with the user, mapping strategy and whether external knowledge is used. Eleven tools were reviewed and summarised by Kalfoglou & Schorlemmer [Kalfoglou & Schorlemmer, 2003] which provided an analysis based on categories such as frameworks, surveys, examples, methods and tools. A comprehensive review on state of the art matching tools is presented in [Euzenat & Shvaiko, 2007 p.153] where comparisons are made based on the types of techniques (discussed earlier in this section) used in them. For a complete list of the tools mentioned above, see appendix B. Given such a large and diverse collection of MOM tools, it can be difficult to determine the right tools for a particular mapping need. In an effort to evaluate matching tools and systems, the OAEI has been organising contests and publishing results on an annual-basis since 2004. The OAEI contest examines the participating tools in a range of mapping tasks across several domains of interest. Though this continuous effort to improve MOM techniques is being made, it is however, difficult to identify a MOM tool that is a clear success based on the OAEI results for reasons discussed next.

First of all, data sets introduced every year differ from those used in previous years. Secondly, the tools that participate in the contests vary each year (e.g. OAEI 2010 contains 4 tracks and 6 data sets with 15 participants [Euzenat et al., 2010]; OAEI 2009 contains 5 tracks and 11 data sets with 16 participants [Euzenat et al., 2009]; OAEI 2008 contains 4 tracks and 8 data sets with 13 participants [Caracciolo et al., 2008]; OAEI 2007 contains 4 tracks and 7 data sets with 18 participants [Euzenat et al., 2007]; OAEI 2006 contains 4 tracks and 6 data sets with 10 participants [Euzenat et al., 2006]; OAEI 2005 contains 3 untracked data sets with 7 participants [Euzenat et al., 2005]; OAEI 2004 contains 1 data set and 10 participants [Euzenat, 2004]). These changing data sets and participants show that the evaluation results generated are from a different sample population each year and in an inconsistent environment. A third factor contributing to the difficulty of identifying the best available MOM tool is that not all test cases are completed by all participants in that year. These variables make aggregated comparison across MOM tools rather difficult. As a result, comparisons of participants in OAEI contests are often made in the context of specific test cases. Given the reasons above, it is difficult to identify a clear winner. However, it is evident that there is a vast array of MOM techniques currently available, which this thesis aims to build upon in the process of achieving CLOM. In particular, the Alignment API offering eight matching techniques is used in this thesis which is discussed next.

2.5.2. The Alignment API

As discussed in the research question (chapter 1, section 1.3), this thesis builds upon resources that are already available in the field of MOM and investigates how these techniques can be facilitated and incorporated in the process of achieving CLOM. In theory, any MOM tool that generates correspondences between formally defined ontologies can be incorporated in the SOCOM and SOCOM++ system. However, rather than seeking and applying the best MOM *tool* (which is difficult to identify), it is the interest of this thesis to investigate how different matching *techniques* (i.e. element-level, structure-level matching strategies) can be supported to achieve CLOM. Thus, the Alignment API[33] is implemented in the CLOM systems presented in this thesis as it offers a range of matching techniques. In particular, eight algorithms are offered by the Alignment API, which include the *NameAndPropertyAlignment* algorithm, the *StrucSubsDistAlignment* algorithm, the *ClassStructAlignment* algorithm, the *NameEq-Alignment* algorithm, the *SMOANameAlignment* algorithm, the *SubsDistName-Alignment* algorithm, the *EditDistNameAlignment* algorithm, the *StringDistAlignment* algorithm. A summary of their functions are presented in table 2-2, based on descriptions presented in the Java documents[34] released with Alignment API version 3.6. The first three algorithms presented in table 2-2 can be categorised as structure-level techniques, which build upon element-level matching techniques and take ontology structures into consideration when generating correspondences. The remaining five algorithms presented in table 2-2 can be categorised as element-level matching techniques, whereby string-based techniques are used to generate correspondences independently from the ontology structures. Because these algorithms offer a good representation of matching techniques that are at the element-level and the structure-level, this Alignment API is chosen to be incorporated by the SOCOM and SOCOM++ system in this thesis[35].

---

[33] http://alignapi.gforge.inria.fr

[34] A list of released APIs and javadocs can be found at https://gforge.inria.fr/frs/?group_id=117&release_id=4104

[35] For a list of other systems that integrates the Alignment API, see http://alignapi.gforge.inria.fr/impl.html

Table 2-2. Matching Algorithms in the Alignment API

Matching Algorithm	Function
NameAndPropertyAlignment	Compares resources based on their names and properties declared
StrucSubsDistAlignment	Compares resources based on substring distance of their names and aggregated these distances with property differences
ClassStructAlignment	Compares resources and the structures they contain
NameEqAlignment	Compares the equality of ontological resource names
SMOANameAlignment	Compares resources using edit distance measures
SubsDistNameAlignment	Compares resources using substring distance on names and properties
EditDistNameAlignment	Compares ontological resource names using Levenshtein distance [Levenshtein, 1966]
StringDistAlignment	Compares ontological resource names regardless of the resource type (i.e. class, property, individual)

An example of the matches generated by the Alignment API using the *SMOANameAlignment* algorithm in the Alignment format[36] is shown in figure 2-3. Each pair of matches (stored in the `<Cell>` element, where the first entity is contained in `<entity1>` and its correspondence is contained in `<entity2>`) generated is accompanied by a confidence level (stored in the `<measure>` element) that ranges between 0.0 (not confident) and 1.0 (confident). For a more detailed overview of the Alignment API, see [Euzenat & Shvaiko, 2007 p.239].

```
…
<map>
    <Cell>
      <entity1 rdf:resource='http://kdeg.cs.tcd.ie/CSWRC/translated#Lecturer'/>
      <entity2 rdf:resource='http://annotation.semanticweb.org/2004/iswc#Lecturer'/>
      <relation>=</relation>
      <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1.0</measure>
    </Cell>
</map>
<map>
    <Cell>
      <entity1 rdf:resource='http://kdeg.cs.tcd.ie/CSWRC/translated#Pages'/>
      <entity2 rdf:resource='http://annotation.semanticweb.org/2004/iswc#homepage'/>
      <relation>=</relation>
      <measure
rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.7481684981684982</measure>
    </Cell>
</map>
<map>
…
```

Figure 2-3. An Example Output from the Alignment API

2.6. Machine Translation

This section presents a brief background on machine translation techniques.

Machine translation is a well-researched field of study that has evolved tremendously over the years since its proposal in 1947 [Weaver & Wiener, 1947]. A brief history of MT is presented by Hutchins [Hutchins, 2004a] that documents the major trends in MT in recent years. Hutchins presents a summary of translation techniques before the 1990s that include direct, interlingua and transfer (which are now

[36] http://alignapi.gforge.inria.fr/format.html

known as rule-based translation systems) in [Hutchins, 2004a], and the more recent techniques developed since the 1990s including example-based MT, statistical MT, hybrid approaches and spoken language MT in [Hutchins, 2004b]. The main concepts behind these techniques are summarised as the following (for a more detailed overview, see [Way, 2010]):

- Direct translation techniques are often designed specifically for one particular pair of natural languages (the source natural language and the target natural language) with minimal syntactic or semantic analysis in the process of translating the source language to the target language.
- Interlingua translation techniques involve a medium between the source and the target language, whereby the source language is converted to interlingua (which can be artificial languages or logics) and then matched from interlingua to the target language.
- Transfer techniques convert both the source and the target language in abstract models, and complete the translation in three steps: convert source language into abstract forms; converts these abstract forms to other abstract representations which originated from the target language; finally convert these abstracts (now oriented by the target language) to the target language.
- Example-based techniques match the source language to previously translated examples in order to determine its translation in the target language.
- Statistical machine translation (SMT) techniques use parallel corpora and compute the probabilities of one word in the source language corresponding with another word in the target language. Types of SMT include word-based [Och & Ney, 2000] and phrase-based [Marcu & Wong, 2002; Koehn et al., 2003].
- Hybrid MT systems combine the above techniques for translations that may be more suited for particular techniques depending on the specific case.

State of the art research in MT is currently led by statistical-based approaches, as seen with major system providers such as Google and Microsoft. Services such as the online GoogleTranslate site[37] and the MicrosoftTranslator site[38] provide free translation services to the general public for small scaled and open-domain requests. These

[37] http://translate.google.com/#
[38] http://www.microsofttranslator.com/

services meet the requirements for the ontology label translation requests made during this thesis research, as they cover the domains explored in the experiments during this research. APIs that are also provided by these services are integrated in the experiments presented in this thesis, as these tools are freely available and use leading SMT techniques which are the state of the art in MT.

2.7. Evaluation Metrics for Ontology Mapping

As this thesis is concerned with improving CLOM quality, the evaluations undertaken thus apply metrics that are used in the state of the art in ontology mapping evaluation. It should be noted that as SOCOM and SOCOM++ presented this thesis are not concerned with ontology localisation, evaluations that are concerned with localisation outcome (e.g. the BLEU score[39] that is often used to measure the quality of the translations generated from a MT system) are not conducted. As already discussed (in section 2.4.2), the requirement for translation in the context of localisation differs from the requirement of translation in the context of CLOM. In the former, a good translation is one that is able to express an equivalent meaning in the culture of the target community. Whereas in the latter, a good translation is one that leads the subsequent MOM techniques to a correct mapping (given such a correct mapping exists in the given scenario). Given the reasons above, evaluations in this thesis apply measures used in the field of ontology mapping which include precision, recall and f-measure. Also, mean and standard deviation are used to evaluate confidence levels. In addition, statistical tests, i.e. two-tailed paired t-tests are used to validate the significance of the findings.

Other ontology mapping evaluation approaches such as goal-oriented approach for ontology mapping is discussed in [Noy & Musen, 2002b; Hollink et al., 2008]. Noy & Musen argue the evaluation of ontology mapping tools should be user-centric and focus on how well a particular task is performed with the assistance of the tool. Hollink et al. propose an end-to-end evaluation approach whereby evaluations are carried out on the performance of the applications that consume the mappings produced by the matching tools. These two approaches focus on how well a particular goal is achieved through the usage of mappings. Though these are sound approaches, they can be difficult to exercise in practice. Systems and applications are often built with specific

[39] The BLEU score [Papineni et al., 2002] aims to evaluate machine-generated translations against that of human-generated translations. It ranges between 0.0 (not close to the human translation) and 1.0 (same with the human translation).

goals in mind. To measure how well these goals are met can involve a range of tests and studies over a period of time, which can be an costly process. Also results collected from such evaluations can be difficult to compare given that the evaluations have taken place in different application contexts and influenced by user subjectivity.

The remainder of this section is organised as follows. Section 2.7.1 presents a tutorial on precision, recall, f-measure, mean, standard deviation and paired t-test. Section 2.7.2 discusses the rationale for using the specific metrics in this thesis.

2.7.1. A Tutorial on Evaluation Metrics

This section presents some background knowledge on precision, recall, f-measure, mean, standard deviation and paired t-test. Section 2.7.1.1 discusses precision, recall and f-measure. Section 2.7.1.2 discusses paired t-test. Section 2.7.1.3 discusses mean and standard deviation.

2.7.1.1. Precision, Recall and F-Measure

Originating from the field of information retrieval (IR) [van Rigsbergen, 1975], precision and recall are first introduced into mapping evaluation in [Do et al., 2002]. In the context of mapping evaluation, precision and recall can be understood as the following. Given the gold standard with *R* results[40] and a set of matches with *X* results, among which *N* of them are correct according to the standard, then

$$\text{Precision} = {}^{N}\!/_{X} \tag{2.1}$$

$$\text{Recall} = {}^{N}\!/_{R} \tag{2.2}$$

This is illustrated by figure 2-4. The gold standard *R* is represented by the gold circle and the set of matches to be evaluated *X* is represented by the purple circle. What they have in common is the correct matches *N*. Precision therefore is a measurement of correctness, and recall is a measurement of completeness. Both precision and recall range between the value of 0.0 and 1.0, where the lower the value, the poorer the correctness or completeness.

[40] In this context, a *result* is a pair of matched entities E_1 and E_2 where E_1 is defined in the source ontology O_1 and E_2 is defined in the target ontology O_2.

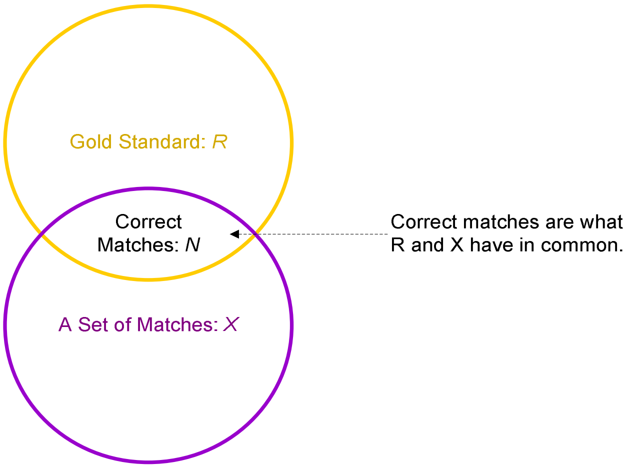


Figure 2-4. Precision & Recall in Mapping Evaluation

It is important to note that precision and recall each accounts for only one aspect of the matching quality (i.e. either correctness or completeness), neither of them alone is an accurate measurement of the matching quality. Precision can be increased at the expenses of recall, or vice versa. For example, table 2-3 shows two scenarios. In scenario i, given a total of 10 matches where all of them are correct, the precision yields 1.00, however, with a total of 100 matches included in the gold standard *R*, the recall is only 0.10. In scenario ii, given a total of 100 matches to be evaluated and a gold standard *R* of 100 matches, the recall is 1.00. However, only 10 matches in *X* are correct, which leads to a low precision of 0.10. These examples demonstrate the importance of evaluating the overall quality of the matches generated which can take both precision and recall into account. To address this issue, f-measure (which too ranges between 0.0 and 1.0) is commonly used in mapping evaluations to illustrate the overall quality of matches, which is computed as:

$$\text{F-Measure} = {2 \times precision \times recall} \Big/ {(precision + recall)} \qquad (2.3)$$

Given the f-measure, a much improved overview on the matching quality (i.e. considers both the correctness and completeness) is thus available. For instance, in the examples shown in table 2-3, both scenarios yield 0.1818 f-measure scores when taken both precision and recall into account.

Table 2-3. Examples of Precision, Recall and F-Measure

Scenario	R	X	N	Precision	Recall	F-Measure
i	100	10	10	1.00	0.10	0.1818
ii	100	100	10	0.10	1.00	0.1818

2.7.1.2. Paired T-Test

The paired t-test is used in hypothesis testing which involves comparisons made upon two related populations. Paired t-tests are often used when "analysing differences between twins, differences in before-and-after measurements on the same subject and differences between two treatments given to the same subject" [Minitab StatGuide, 2007]. For example, the following scenario can be tested with paired t-test: a group of hypertension patients have been treated with a new drug over a period of time, their blood pressures are recorded before and after the treatment. Let *x* be the records before the treatment and *y* be the records after the treatment, it is the difference between *x* and *y* that is of interest and examined in paired t-test. As *x* and *y* are collected from the same group of people, we say *x* and *y* are related or paired. Weiss [Weiss, 2010] define the following procedures in a paired t-test for two population means μ_1 and μ_2 as:

> "*Step 1: the null hypothesis is* H_0*:* $\mu_1 = \mu_2$*, and the alternative hypothesis is* H_a*:* $\mu_1 \neq \mu_2$ *(two tailed) or* H_a*:* $\mu_1 < \mu_2$ *(left tailed) or* H_a*:* $\mu_1 > \mu_2$ *(right tailed);*
>
> *Step 2: decide on the significance level* α*;*
>
> *Step 3: compute the value of the test statistic* $t = \dfrac{\bar{d}}{S_d/\sqrt{n}}$ *and denote value* t_0*.*
>
> *Step 4: the t-statistic has* $df = n-1$*, compute p-value;*
>
> *Step 5: if* $p \le \alpha$*, reject* H_0*; otherwise, do not reject* H_0*.*"

Figure 2-5 illustrates the two tailed (figure 2-5-a), left tailed (figure 2-5-b) and right tailed (figure 2-5-c) paired t-tests. Paired t-tests are interested in the difference between two samples, in the case of one tailed t-tests (left tailed or right tailed) the direction of the difference (either $\mu_1 < \mu_2$ or $\mu_1 > \mu_2$) is examined. In the case of a two tailed t-test, the particular direction of the difference is not of concern, but rather establishing whether a difference exists between two samples or these samples are in fact from the same population.

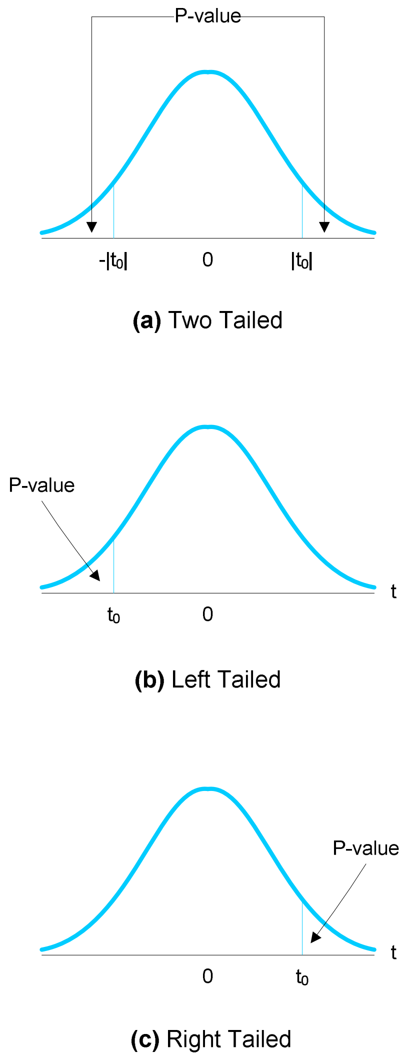


Figure 2-5. Two Tailed and One Tailed Paired T-Test [Weiss, 2010 p.481]

The significance level α ranges between 0.0 and 1.0, and is the maximum acceptable level of risk for rejecting the null hypothesis. The most commonly used α level is 0.05, whereby the chance of finding an effect that does not exist is only 5%. The reason for this being:

> "*The value for which P=0.05, or 1 in 20, is 1.96 or nearly 2; it is convenient to take this point as a limit in judging whether a deviation ought to be considered significant or not. Deviations exceeding twice the standard deviation are thus formally regarded as significant. Using this criterion we should be led to follow up a false indication only once in 22 trials, even if the statistics were the only guide available. Small effects will still escape notice if the data are insufficiently numerous to bring them out, but no lowering of the standard of significance would meet this difficulty.*" [Fisher, 1958 p.44]

The p-value ranges between 0.0 and 1.0 and is used to determine the appropriateness of rejecting the null hypothesis. The smaller it is, the smaller the probability that rejecting the null hypothesis is a mistake. If the p-value is less than or equal to the α level, it can be said that there is good evidence against the null hypothesis; if it is greater than the α level, it can be said that there is not enough good evidence to reject the null hypothesis.

2.7.1.3. Mean and Standard Deviation

As mentioned in section 2.5.2 (figure 2-3), the matches generated are accompanied by confidence levels that range between 0.0 and 1.0. These levels are generated by the Alignment API and are used to indicate the tool's confidence in a match made, where the higher they are, the more confident the matches are. To evaluate these confidence measures, mean and standard deviation are calculated. This section discusses mean and standard deviation in detail.

Given a set of matches X and their accompanying confidence levels, mean is the average confidence level found in the correct matches N. It is a measure of centre (i.e. most typical value of a data set), and is calculated as the sum of all confidence levels divided by the sum of matches. In other words, the confidence mean is simply the average confidence level found in a set of correct matches N. The higher the mean, the more confident are the matches. In this thesis, means are calculated in the evaluation to indicate one aspect of the matching quality: the confidence of the matches generated.

Standard deviation is a measure of variation. It indicates how far, on average, the observations (in this case, the confidence levels) are from the mean. For a data set with a large amount of variation, the observations will on average be far from the mean, which implies that the standard deviations will be large. Similarly, for a data set with a small amount of variation, the observations will on average be close to the mean, indicated by a small standard deviation.

2.7.2. Evaluation Metrics used in This Thesis

This section presents the rationale for applying the evaluation metrics discussed in section 2.7.1 in this thesis.

In recent years, using metrics that originated from the field of information retrieval (IR) such as precision, recall and f-measure have become widely adopted as a less expensive and more comparable means of evaluating ontology mapping systems, which focuses on evaluating the general functionality rather than evaluating the system in particular application contexts. This approach measures a set of machine-generated mappings against a gold standard that had been generated by human experts. The quality of the machine-generated mappings is measured as how closely (via precision, recall and f-measure scores, discussed in section 2.7.1.1) the mappings correspond to those gold standard that had been generated by humans independent of application context. The closer the mappings are to the gold standard, the higher the quality.

This IR-inspired evaluation approach is recommended by [Euzenat & Shvaiko, 2007 Chapter 7 p.193]. Euzenat & Shvaiko point out that the evaluation procedure should be a reproducible and continuous process with pre-defined rules and published results that include not only the evaluation results but also the actual mappings themselves. Guided by this principle, in practice, a widely accepted approach (that has been enforced in the OAEI contests since 2004) is to use a benchmark in the ontology mapping evaluation, which is considered as the gold standard of mappings between a particular ontology pair. Evaluations of mappings generated by other systems are then compared against this gold standard. A gold standard is "used repeatedly for (i) testing the improvement or degradation of a system with certainty, (ii) situating a system among others" [Euzenat & Shvaiko, 2007 p.194]. This evaluation approach using gold standards is thus adopted in this thesis for their comparable nature and ease of use.

When calculating f-measure, weights can be assigned to precision and recall to illustrate their perceived importance. Do et al. [Do et al., 2002] define the weighted f-measure as $F\text{-}Measure = \frac{precision \times recall}{(1-a)\times precision + a \times recall}$ where $0 \le \alpha \le 1$. When α=1, no importance is assigned to recall; when α=0, no importance is assigned to precision. The higher the α, the more importance is given to precision. This use of weighted f-measure is demonstrated by Kaza & Chen [Kaza & Chen, 2007], where precision is considered twice as important as recall (i.e. $\alpha=0.\overline{6}$). However, a value of 0.5 is commonly assigned to the weight α (as seen in OAEI contests[41]), so that precision and recall are considered equally as important as each other. In other words, when α=0.5:

[41] OAEI 2010 results can be found at http://oaei.ontologymatching.org/2010/results/oaei2010.pdf; OAEI 2009 results can be found at http://oaei.ontologymatching.org/2009/results/oaei2009.pdf; OAEI 2008

F-Measure = $\frac{precision \times recall}{(1-a) \times precision + a \times recall} = 2 \times precision \times recall \Big/ (precision + recall)$

as shown in formula (2.3) previously (see section 2.7.1.1). This thesis uses the formulas (2.1), (2.2) and (2.3) where precision and recall have equal weight in the evaluations.

Melnik et al. [Melnik et al., 2002] introduce *overall* as a measurement for the effort required to correct the errors in a set of mappings, which is defined as *Overall* = $recall \times \left(2 - 1 \big/ precision\right)$. Overall scores range between the value of -1.0 and 1.0, and are always lower than f-measure scores. Unlike precision, recall and f-measure, overall is not commonly used in the OAEI contests. As this thesis is concerned with the quality of the mappings and measuring improvement (if there is any) in the matching quality (through the values of precision, recall and f-measure), but not quantifying the post-mapping editing efforts involved, overall scores are not generated in this thesis.

Given a gold standard *R*, a set of matches *X* with *N* correct matches, the *fallout* can also be calculated as: *Fallout* = $(X - N) \big/ X$. Fallout quantifies the incorrect matches found in a set of matches *X*, and ranges between 0.0 and 1.0 where the higher it is, the more incorrect matches there are in *X*. Since *precision* + *fallout* = $N \big/ X + (X - N) \big/ X = 1$, in other words, fallout = 1 – precision, where the higher the fallout the lower the precision and vice versa, in this thesis, fallout is considered to be redundant data since it does not offer more insight into the mapping quality and thus is not generated.

Precision and recall have been criticised in [Ehrig & Euzenat, 2005; Euzenat, 2007] for (1) their inability to distinguish matches that are almost correct and those that are completely wrong, (2) as well as their limitations to evaluate narrow-broad and broad-narrow matches. To address these shortcomings, generalised (aimed to improve the first limitation) and semantic precision and recall (aimed to improve the second limitation) have been proposed. Generalised precision and recall are proposed in [Ehrig & Euzenat, 2005], where the proximity of ω(X, R) is measured instead of strictly looking for the overlap $|X \cap R|$. However, this thesis does not differentiate matches that are almost correct from matches that are complete misses. They are viewed as two

---

results can be found at http://oaei.ontologymatching.org/2008/results/oaei2008v11.pdf; OAEI 2007 results can be found at http://oaei.ontologymatching.org/doc/Proceedings-OM-2007.pdf; OAEI 2006 results can be found at http://oaei.ontologymatching.org/doc/Proceedings-OM-2006.pdf; OAEI 2005 results can be found at http://oaei.ontologymatching.org/doc/intont2005proceedings.pdf; OAEI 2004 results can be found at http://oaei.ontologymatching.org/2004/Contest/results facets of the same type - being incorrect matches. Although one could argue that an almost correct match may require less effort to correct than a match that is a total miss. However, they are nevertheless both incorrect and require further correction. Whether a great deal of effort or less is involved is not of concern. In other words, it is not of interest to measure the extent of their incorrectness, thus generalised precision and recall are not applied in this thesis. Semantic precision and recall are proposed in [Euzenat, 2007], where narrow-broad (e.g. book ≤ publication) and broad-narrow (e.g. publication ≥ book) matches are taken into account in the evaluation in addition to equal-equal (e.g. lecturer = lecturer) matches. In this thesis, however, the matching tool (i.e. the Alignment API discussed in section 2.5.2) incorporated into the CLOM systems only generate equal-equal matches in the experiments, since narrow-broad and broad-narrow matches do not exist, semantic precision and recall are thus not applied in the evaluation used in this thesis.

In addition to comparing precision, recall and f-measure, this thesis applies two tailed paired t-tests to test the statistical difference between the proposed CLOM approach and the baseline approach. A working example is presented next. Given a pair of ontologies that are labelled in different natural languages O_1 and O_2, O_1 is mapped to O_2 using two CLOM systems: the baseline system and the SOCOM system, generating mappings M_1 and M_2 respectively. Eight different matching algorithms are applied in the mapping process, which lead to eight sets of matches in M_1 and M_2 each. Based on the gold standard, M_1 and M_2 are evaluated using precision, recall and f-measure. The f-measure scores are considered as indicators of the overall matching quality (since they take both precision and recall scores into account), and the f-measure generated from M_1 and the f-measure generated from M_2 are compared. Since these f-measure scores are generated from mappings conducted on the same ontology pair using the same set of matching algorithms, they are therefore paired with each other. To test whether a difference exists between the overall quality found in M_1 and M_2, two-tailed paired t-test is carried out on the f-measure scores. The null hypothesis is:

H_0: $M_1 = M_2$ (there is no difference between the matching quality in M_1 and M_2);

And the alternative hypothesis is:

H_a: $M_1 \neq M_2$ (there is a difference between the matching quality in M_1 and M_2);

Table 2-4 contains results taken from the evaluation of SOCOM++ in trial five (discussed in chapter 5, section 5.4.3.2, experiment one). Using Minitab 15[42] (all paired t-tests shown in this thesis are carried out using Minitab), at the 5% significance level, paired t-test results are computed and are shown in figure 2-6. As shown in figure 2-6, the t-value generated is -3.40 (using the formula earlier from [Weiss, 2010]) which corresponds to a p-value of 0.011. The t-test also shows that a 95% confidence interval for the difference between M_1 and M_2 is from -0.2170 to -0.0389.

Table 2-4. Paired T-Test on F-Measure

Matching Technique	F-Measure	
	M_1	M_2
1 NameAndPropertyAlignment	0.3297	0.3509
2 StrucSubsDistAlignment	0.3244	0.3882
3 ClassStructAlignment	0.3244	0.4025
4 NameEqAlignment	0.5021	0.7797
5 SMOANameAlignment	0.3625	0.4025
6 SubsDistNameAlignment	0.3412	0.4074
7 EditDistNameAilgnment	0.3391	0.5385
8 StringDistAlignment	0.5021	0.7797

```
Paired T for M1 - M2

             N      Mean    StDev   SE Mean
M1           8    0.3782   0.0774    0.0274
M2           8    0.5062   0.1772    0.0627
Difference   8   -0.1280   0.1065    0.0377


95% CI for mean difference: (-0.2170, -0.0389)
T-Test of mean difference = 0 (vs not = 0): T-Value = -3.40  P-Value = 0.011
```

Figure 2-6. Paired T-Test for M_1 and M_2

The direction of the difference between M_1 and M_2 is known since we already know higher f-measure indicates higher matching quality. Thus two-tailed paired t-tests are applied in this thesis, since it is our interest to find out whether the f-measure scores collected from two systems are from two different populations or indeed from the same population. Using the hypothesis test such as the paired t-test adds statistical power to the findings concluded from the evaluation, which helps the author to conclude with confidence in this thesis.

In this thesis, standard deviations are used to indicate the dispersion of confidence levels found in a set of correct matches. The higher the standard deviations, the more dispersed are the confidence levels. For example, in the evaluation of SOCOM++ trial one (discussed in chapter 5, section 5.4.2.1, experiment one), the confidence mean found in the baseline system is 0.8830 with a standard deviation of 0.1391. This

[42] http://www.minitab.com/en-US/default.aspx

indicates that on average, the values in the data set (i.e. all the confidence levels found in the correct matches that have been generated using the baseline system) tend to differ from the mean by ± 0.1391. In contrast, the mean found in SOCOM is 0.9646 with a standard deviation of 0.0613. This indicates that on average, the values in this data set (i.e. all the confidence levels found in the correct matches that have been generated using SOCOM) tend to differ from the mean by ± 0.0613. In other words, the correct matches found in SOCOM are not only more confident but also less dispersed.

2.8. Summary

This chapter presents related background and a state of the art review on CLOM. A survey of how CLOM is achieved to date is presented in this chapter, whereby a translation-based approach to CLOM is identified as the most advanced work in CLOM. The translation-based approach converts a cross-lingual mapping problem into a monolingual mapping problem, whereby translation techniques such as MT tools are used to overcome natural language barriers and MOM tools are applied subsequently. Related background reviews on MT and MOM are thus also included in this chapter. Finally, evaluation metrics applied in this thesis are introduced and discussed.

Arising from the review on state of the art in CLOM, this thesis asks an important question regarding the current translation-based approach to CLOM: it is shown in the literature that translations can serve as a means to the completion of CLOM, but just how suitable are these translations in the matching sense (i.e. correct mappings are generated) as opposed to the linguistic sense (i.e. correctly localised)? This question is investigated next.

3 THE CHALLENGE OF TRANSLATION IN CROSS-LINGUAL ONTOLOGY MAPPING

3.1. Chapter Overview

This chapter presents the building process behind the appropriate ontology label translation (AOLT) concept upon which this research is grounded. In particular, the baseline approach to CLOM as identified in the state of the art (discussed in chapter 2, section 2.4) is examined in two experiments involving ontologies labelled in Chinese and English. The effectiveness of the baseline approach is investigated and the findings from the evaluation motivated and inspired the creation and the development of the AOLT process. In particular, the experiments aim to identify the limitations and challenges faced by the baseline approach to CLOM, which this thesis aims to address.

This chapter is organised as follows. The motivation of the experiments presented in this chapter is discussed in section 3.2. An overview of the baseline approach, including its architecture and an implementation are presented and discussed in section 3.3. Two experiments designed to investigate the effectiveness of the integrated baseline system in CLOM scenarios, together with their experimental setup, findings and conclusions are discussed in section 3.4. Finally, section 3.5 presents a summary of this chapter. The baseline system to CLOM (presented in section 3.3), the two experiments and their evaluation results (presented in section 3.4) have been published in the paper titled *Cross-Lingual Ontology Mapping - An Investigation of the Impact of Machine Translation*, at the 4th Annual Asian Semantic Web Conference (ASWC 2009), LNCS 5926, pp. 1-15, in December 2009.

3.2. Experimental Motivation

As discussed in chapter 2, a popular approach to CLOM is to use MT techniques to turn a cross-lingual mapping problem into a monolingual mapping problem first which can then be solved by existing MOM tools next. However, the quality of the translated resource labels and the impact of the translation process on the mappings subsequently generated using this approach has not yet been investigated.

The goal of the experiments presented in this chapter is to investigate how the translation of labels may affect the mapping quality. In this chapter, emphasis is placed on the quality of the machine translated ontology labels and how they may impact on the effectiveness of the baseline approach to CLOM. The evaluation of the baseline implementation is composed of two experiments. The first experiment investigates the quality of the machine translated resource labels and how appropriate they are in the given mapping context. The second experiment investigates the quality of the mappings that were generated using the baseline system.

3.3. The Baseline Approach

The review presented in the previous chapter (section 2.4) has identified the baseline approach as the current state of the art in CLOM. This approach uses translation as a means to convert a cross-lingual mapping problem into a monolingual mapping problem, which is then solved by MOM tools. However, little attention has been paid to the effectiveness of this approach. More specifically, the quality of the mappings generated using such an approach have not yet been evaluated. To investigate this matter further, an implementation of the baseline approach to CLOM is examined through two experiments in this chapter. The architecture of the baseline approach is outlined in section 3.3.1. The technologies used to implement this baseline system are discussed in section 3.3.2.

3.3.1. Architecture Overview

As identified in chapter 2 (section 2.4), the baseline approach employs a two-tier strategy to achieve CLOM. First, resource labels in one ontology are translated into the natural language used by the other ontology(ies). Secondly, monolingual ontology matching techniques are applied to generate matches. The workflow of this approach can be illustrated by figure 3-1.

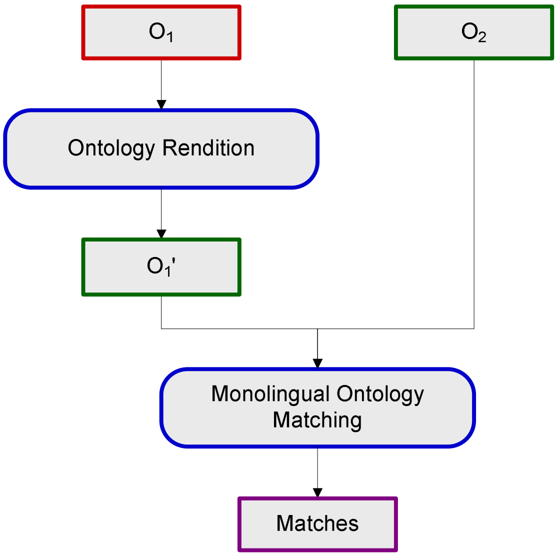


Figure 3-1. An Architecture of the Baseline Approach to CLOM

Given ontologies O_1 and O_2 that are labelled in different natural languages, O_1 is first transformed to O_1' through the *ontology rendition* process, so that O_1' contains the same semantics as O_1 except its resources are labelled in the target natural language used by O_2. Ontology rendition can be defined as a process in the ontology development that consists of two roles, converting and interpreting [Zhao et al., 2003]. The converting role is the transformation of an ontology where the output has "formally different but theoretically equivalent" semantics, e.g. translating ontologies from OWL to RDF via Web-PDDL [Dou et al., 2004]. The interpreting role renders formally specified commitments, which is the aim of the ontology rendition shown in figure 3-1. More specifically, the same semantics can be found in O_1' as one would find in O_1. In addition, these semantics are defined using the same formal language (i.e. RDF, OWL etc.). The difference between O_1 and O_1' is the natural language of the labels used by their respective resources. In contrast to O_1, the labels of the resources in O_1' are labelled in the natural language used by O_2.

An example of the input and the output from ontology rendition is shown in figure 3-2. In this example, the source ontology: O_1 is labelled in English and the target ontology: O_2 is labelled in Chinese. The rendered ontology: O_1' is thus labelled in Chinese. Note that new namespace declarations are assigned to resources in the rendered ontology. This is because the base URI is the unique identifier for an ontology and the resources within, which means that the identifiers in O_1' should not point to the original resources in O_1. As discussed in chapter 1 (section 1.2), the same resource (with one unique identifier) can have multiple tags (such as the use of `<rdfs:label>` in figure 1-1) that illustrate the given ontology label in various natural languages. However, as O_1' needs to be self-contained (i.e. a formal ontology on its own that can be processed by machines) so that it can be matched to O_2, the resources within need to be well-formed (i.e. resources in O_1' need to have unique identifiers that are not to be confused with the resources in O_1). Therefore, as shown in figure 3-2, new namespace declarations are assigned to the translated labels in the rendered ontology. This ontology rendition process is necessary in order to apply existing MOM techniques.

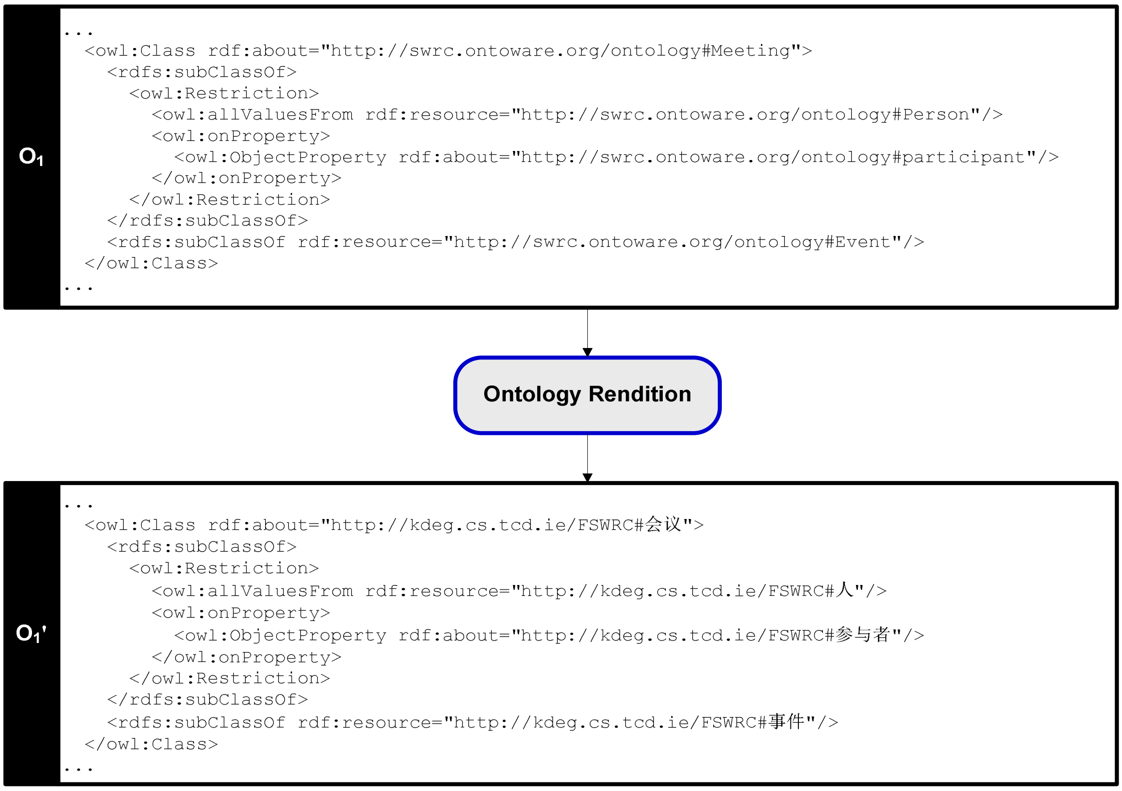


Figure 3-2. An Example of Ontology Rendition

Given that O_1' is in the same natural language as O_2, a range of existing MOM techniques (discussed in chapter 2, section 2.4) can be applied to generate matches between O_1' and O_2 via the *monolingual ontology matching* process. These matches are considered as correspondences between O_1 and O_2 as O_1' contains the same semantics as O_1. To investigate the effectiveness of this baseline approach, an implementation of it is developed using off-the-shelf MT and MOM tools. This is discussed next.

3.3.2. Implementation

A Java implementation of the baseline approach to CLOM is developed. The Java code for the baseline system can be found at `root/Baseline/src/`, and the Jar files required to run the system can be found at `root/Baseline/bin/` on the DVD. Figure 3-3 presents a deployment diagram that shows the components of the implemented system and how they are related. The dashed boxes outline the two main steps of the baseline system, namely the ontology rendition and the ontology matching process.

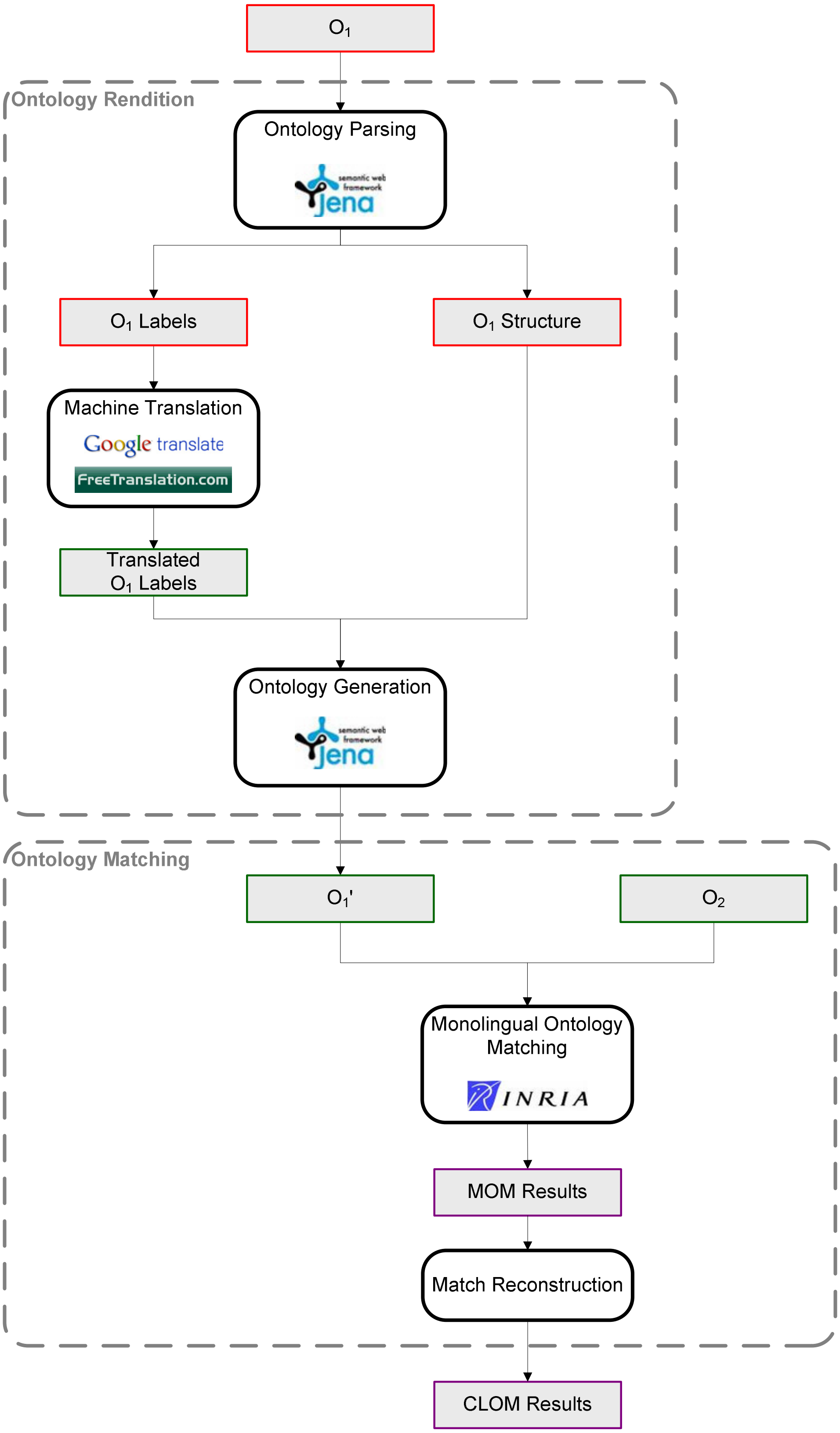


Figure 3-3. An Implementation of the Baseline Approach to CLOM - Deployment Diagram

To render O_1', labels of the ontological resources (i.e. classes, datatype properties, object properties and individuals) from O_1 are extracted first using the Jena Framework[43], version 2.5.5. These labels are then passed onto the MT tools to generate translations in the target natural language. Two MT tools have been used in the implementation of the baseline system: the GoogleTranslate API[44] version 0.4 provided by Google Inc. and the FreeTranslation[45] online translator which is provided by SDL[46]. These tools are chosen as they are representative of the state of the art techniques in MT (discussed in chapter 2, section 2.6). Given the structure of O_1, the translated resource labels are arranged accordingly to generate O_1' using the Jena Framework. This process can be illustrated by the sequence diagram shown in figure 3-4. Code snippets for this rendition process using the GoogleTranslate API can be found in appendix C, section C.2, figure C-1.

Figure 3-4 shows the lifelines of eleven objects: `JenaFramework`, `OntModel`, `OntClass`, `DatatypeProperty`, `ObjectProperty`, `Individual`, `LabelReconstruction`, `MT`, `AOLTRecord`, `CollisionResolution` and `O1'`. To run the application, the user first sets the input: O_1 (i.e. locate the ontology). Using the *`JenaFramework`*, O_1's *`OntModel`* (an interface supported by the Jena Framework) is generated next which presents a syntax for accessing the data contained in O_1. The application then accesses the declared resources via this `OntModel` of O_1. A copy of this `OntModel` is also generated at this stage which will eventually contain data for O_1'. For a resource *R* in O_1,

- *R*'s label is extracted via the *`OntClass`* (for a class), *`DatatypeProperty`* (for a datatype property), *`ObjectProperty`* (for an object property) or the *`Individual`* (for an individual) interface. These interfaces are supported by the Jena Framework;
- if *R*'s label is concatenated, it needs to be converted in a way that it can be processed by MT tools (i.e. in the label's natural language format, more details are discussed next) via the *`LabelReconstruction`* object;
- Translation for *R* is obtained next from the *`MT`* object;

[43] http://jena.sourceforge.net
[44] http://code.google.com/p/google-api-translate-java Note: The GoogleTranslate API has been officially deprecated as of May 26, 2011, and will be shut off completely on December 1, 2011.
[45] http://www.freetranslation.com
[46] SDL provides information management solutions for its clients. More information can be found at http://www.sdl.com

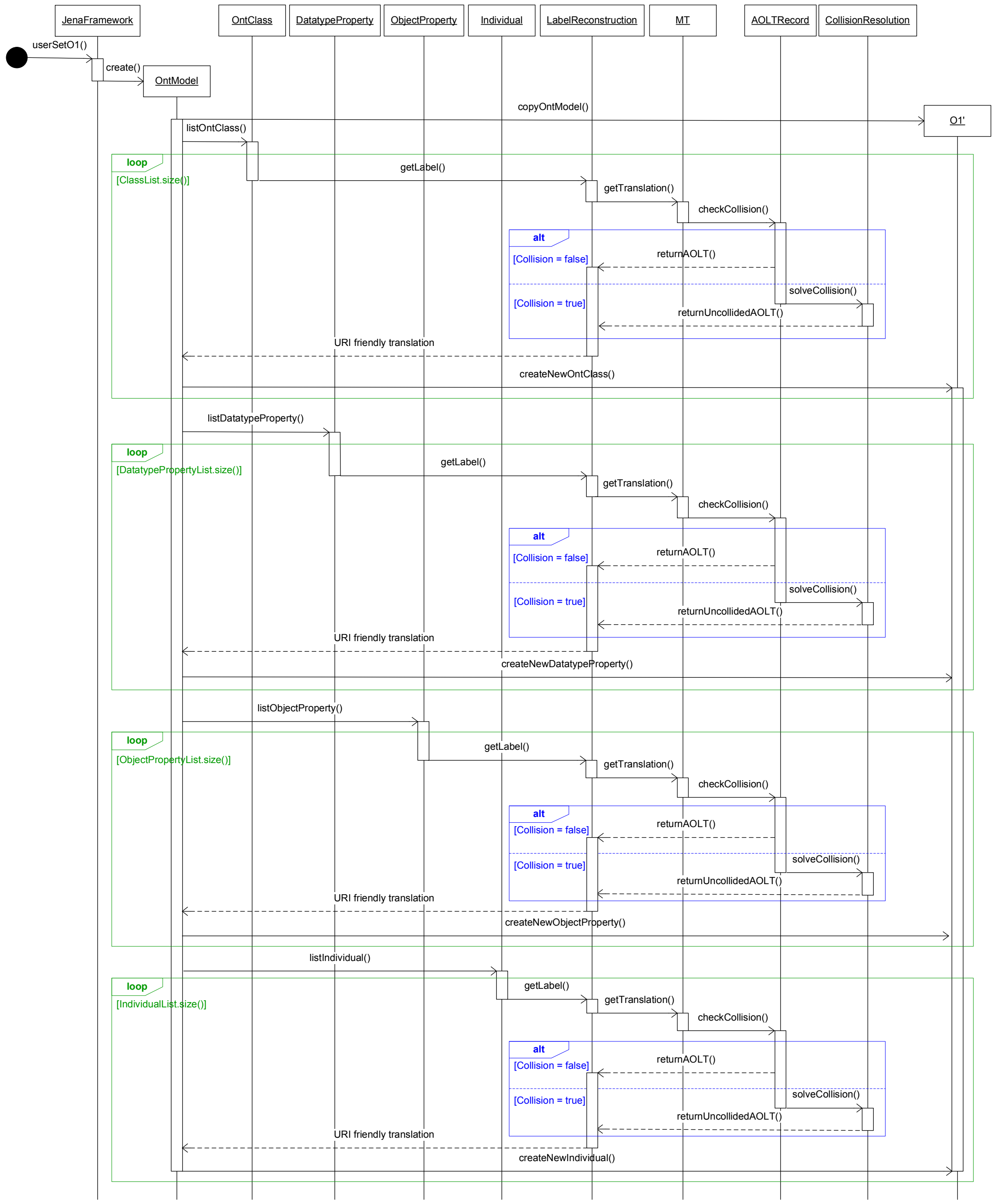


Figure 3-4. Ontology Rendition - UML Sequence Diagram

- This translation is stored in the `AOLTRecord` object. A translation can only be considered as the AOLT result if it is free of collision.

- If a collision is found (i.e. the AOLT result at hand is the same with a previously stored translation in the AOLTRecord), the `CollisionResolution` object is called to resolve collisions (discussed next).

- The collision-free AOLT result is then converted to be URI friendly (i.e. white spaces are removed as they are not allowed in unique resource identifiers) via the `LabelReconstruction` object.

- Finally, a new resource (i.e. a semantic equivalent of *R* that is labelled in the target natural language) is created in O_1' via the `OntModel`.

The process above is repeated for each resource in O_1. Ontology labels are often concatenated to create well-formed Unique Resource Identifiers (URIs) since white spaces are not allowed in the naming convention. For example, a class resource *research project* can be labelled in the ontology as (among others):

```
<owl:Class rdf:about="http://swrc.ontoware.org/ontology#Research_Project"/>
```

or

```
<owl:Class rdf:about="http://swrc.ontoware.org/ontology#ResearchProject"/>
```

As the integrated MT tools cannot process such concatenated labels, these labels are split into sequences of their constituent words in natural language format before being passed to the MT tools, as shown in figure 3-4 as the `LabelReconstruction` lifeline. This is achieved by recognising concatenation patterns. In the first example shown above, underscores are replaced by white spaces. A code snippet of this is presented in appendix C, section C.2, figure C-2. In the second example shown above, white spaces are inserted before each capital letter found other than the first one. A code snippet of this is presented in appendix C, section C.2, figure C-3. Though other ways to concatenate labels are possible, only these two types of concatenations are handled by the implementation. This is because only these types of concatenations exist in the ontologies which this thesis has experimented with. Note that concatenated words differ from compound words or portmanteau words. A compound word consists of two or more free morphemes which are standalone on their own. For example, *football* is a compound word that is composed of *foot* and *ball*, where both *foot* and *ball* are standalone words. A portmanteau word blends parts of two or more words which are not always standalone free morphemes. For example, *brunch* is a portmanteau word that blends *br* from *breakfast* and *unch* from *lunch*, where by neither *br* nor *unch* are standalone words. Neither compound words nor portmanteau words present an issue for the system implementation because they can be translated using the integrated MT tools. In contrast, concatenated words are constructed in a way to comply with ontology resource naming standards. These concatenations present an issue for the baseline system as they are unrecognisable natural language context to the MT tools, which is why they are reconstructed to their constituent words as discussed earlier.

Both integrated MT tools return one and only one translation for a given label at a time, however, translation collisions can happen when a MT tool returns the same result for several labels in O_1. For instance, in the Semantic Web Research Community (SWRC) ontology[47], using the GoogleTranslate API version 0.4, the class *Conference* and the class *Meeting* are both translated to *会议* (meaning "meeting" in Chinese). To resolve such collisions, the baseline system checks whether a translation already exists in the O_1' ontology or not. If so, an integer (that is checked to be free of collision) is assigned to the translated label which is under consideration. In the aforementioned example, as *会议* already exists as the class label: *Conference*'s translation, for the collided class: *Meeting*, its translated label becomes *会议0* in O_1'. This ensures that both resources will have well-formed (i.e. unique) URIs. These numbers are selected at random with the intent of avoiding the introduction of any kind of patterns into the translation selection process. This translation collision issue is not mentioned in any of the translation-based approaches to CLOM to date and it is not clear how collisions are solved in the papers discussed in chapter 2, section 2.4. In the implementation presented in this thesis, adding random numbers to solve collisions is a way to overcome disruptions to the execution of the system. Ideally, human experts are present to resolve collisions. However, this may not always be possible. The baseline implementation shown in this thesis allows the system to automatically resolve collisions without the assistance of a human. Lastly, it should be noted that when structuring the translated labels, white spaces are removed from the translations returned by the MT tools in order to generate well-formed URIs in O_1', as the `LabelReconstruction` timeline illustrates in figure 3-4. Label reconstruction concatenates the translated labels in the same way as the original ontology, i.e. white spaces are either removed or replaced by underscores.

Once the source ontology is labelled in the natural language used by O_2, the Alignment API[48], version 2.5 is applied to generate matches as shown in figure 3-3. The code snippet shown in appendix C, section C.2, figure C-4 demonstrates how the Alignment API is integrated into the baseline system. An example output from the Alignment API can be found in chapter 2, section 2.5.2, figure 2-3. Though two or more algorithms of the Alignment API can be combined to generate matches, however, as it is of interest to investigate how each algorithm behave given the same mapping context, all eight algorithms are executed independently. Knowing the matches between

[47] http://ontoware.org/swrc/swrc/SWRCOWL/swrc_v0.3.owl
[48] http://alignapi.gforge.inria.fr

O_1' and O_2, as well as how the labels in O_1 have been translated (i.e. which resources in O_1' corresponds to the resources in O_1), the match reconstruction process rearranges the MOM matches to finally generate the CLOM matches (between O_1 and O_2). Note that although the diagram in figure 3-3 shows a rendered O_1' that is matched to O_2, the baseline system is applicable to reversed source and target ontology, i.e. O_2 can be rendered to O_2' which can then be matched to O_1.

The baseline system is representative of the current translation-based approach to CLOM (see [Zhang et al., 2008; Wang et al., 2010; Wang et al., 2009; Trojahn, 2010] , this is discussed in chapter 2, section 2.4). Though it may be argued that the matching outcome is conditioned upon the specific MT and MOM tools used in the implementation, however, the typical (i.e. turn a CLOM problem to a MOM problem through translations) process to achieve CLOM in the baseline system is nevertheless representative and thus serves as a reference point for this thesis[49]. The effectiveness of the baseline system is investigated next. Particularly, how the translated ontology labels impact on the mapping quality is examined.

3.4. Experiments

In this section, two experiments are discussed. Experiment one aims to investigate the appropriateness (from the mapping view point) of the translations in the rendered ontology. Experiment two evaluates the quality of the matches generated using the baseline system. The ontologies used in the experiments include the SWRC ontology[50] (in English, developed by Ontoware[51]) and the ISWC ontology[52] (in English, developed by Semantic Web, Annotation & Authoring[53]). These ontologies contain general concepts that are often seen in the research domain. The SWRC ontology contains 54 classes, 30 datatype properties, 44 object properties and no individuals - a total of 128 resources. The ISWC ontology is of a similar size, containing 33 classes, 17 datatype properties, 18 object properties and 50 individuals - a total of 118 resources. Figure 3-5 presents partial views of these ontologies in the Protégé[54] editor. The SWRC and the

[49] In the later chapters of this thesis, evaluations will show that the matching quality can be improved by the proposed AOLT process even though the same MT and MOM tools used in the baseline system are implemented in SOCOM and SOCOM++.
[50] The SWRC ontology can be downloaded at http://ontoware.org/swrc/swrc/SWRCOWL/swrc_v0.3.owl
[51] http://www.ontoware.org/index.html
[52] The ISWC ontology can be downloaded at http://annotation.semanticweb.org/ontologies/iswc.owl
[53] http://annotation.semanticweb.org/portal_url/portal_url
[54] http://protege.stanford.edu/

ISWC ontology are chosen for mainly three reasons. Firstly, they are both developed by third parties (i.e. free of interference from this author). Secondly, they contain overlapping domains and different structures, which are examples of ontologies typically presented in mapping scenarios. Thirdly, the domain of these ontologies is familiar to the author of this thesis, whereby investigations on the appropriateness of the translations can be carried out with ease.

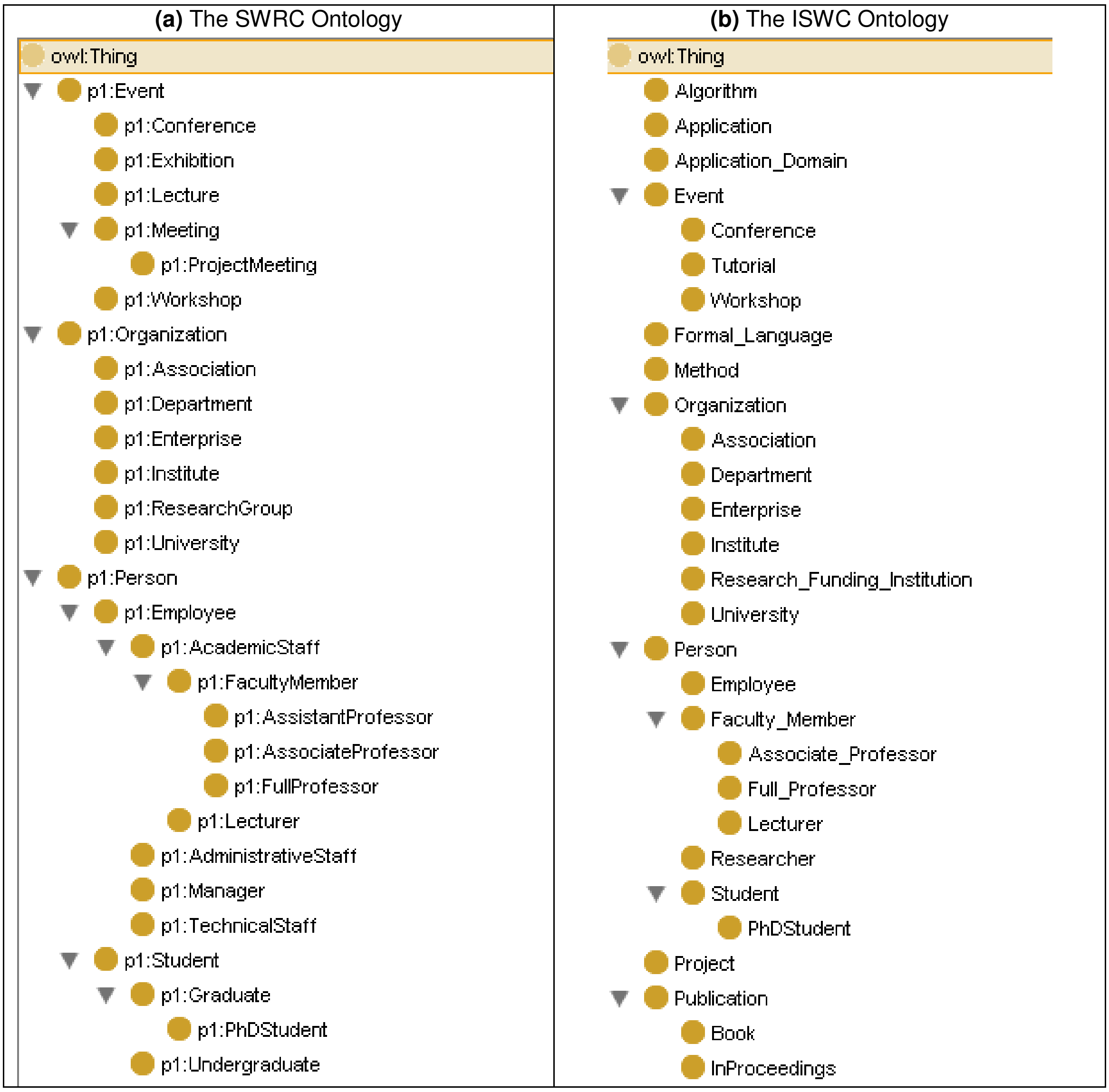


Figure 3-5. Partial Views of the SWRC and the ISWC Ontology in Protégé

The remainder of section 3.4 is organised as follows. Section 3.4.1 presents the experimental setup, the findings and analysis of the first experiment. Section 3.4.2 presents the experimental setup, findings and analysis of the second experiment. Finally, conclusions drawn from the two experiments are presented in section 3.4.3.

For raw data collected from these experiments, see the accompanying DVD:

- The rendered ontologies from experiment one can be found at `root/BaselineExperiments/Experiment1/RenderedOntologies/`

- The matches generated from experiment one can be found at `root/BaselineExperiments/Experiment1/Matches/`
- The evaluation (in spreadsheet format) from experiment one can be found at `root/BaselineExperiments/Experiment1/Evaluation/`
- The rendered ontologies from experiment two can be found at `root/BaselineExperiments/Experiment2/RenderedOntologies/`
- The matches generated from experiment two can be found at `root/BaselineExperiments/Experiment2/Matches/`
- The evaluation (in spreadsheet format) from experiment two can be found at `root/BaselineExperiments/Experiment2/Evaluation/`

3.4.1. Experiment One

Experiment one aims to examine the appropriateness of translations from a mapping point of view during the ontology rendition process. The experimental setup is outlined in section 3.4.1.1, followed by the findings and analysis in section 3.4.1.2.

3.4.1.1. Experimental Setup

The goal of experiment one is to investigate whether there are side effects of translating ontology labels during ontology rendition. Three renditions of the same ontology are generated and then mapped to one another. Assuming appropriate translations are generated for all the labels in the given ontology, then the translated labels in all three renditions should be highly similar. This implies that the mappings from any pair of renditions of the same ontology should therefore be highly similar to one another. Whether this assumption is true or false is examined.

In this experiment, the SWRC ontology is converted from the original English version to its Chinese renditions using two approaches: the baseline system and a human expert (being the author of this thesis). Three versions of the SWRC ontology are created as shown in figure 3-6:

- the FSWRC ontology is generated using the baseline system utilising the FreeTranslation online translator as the MT component;
- the GSWRC ontology is generated using the baseline system utilising the GoogleTranslate API as the MT component;

- and the HSWRC ontology is manually generated by the author of this thesis using the Protégé[55] ontology editor.

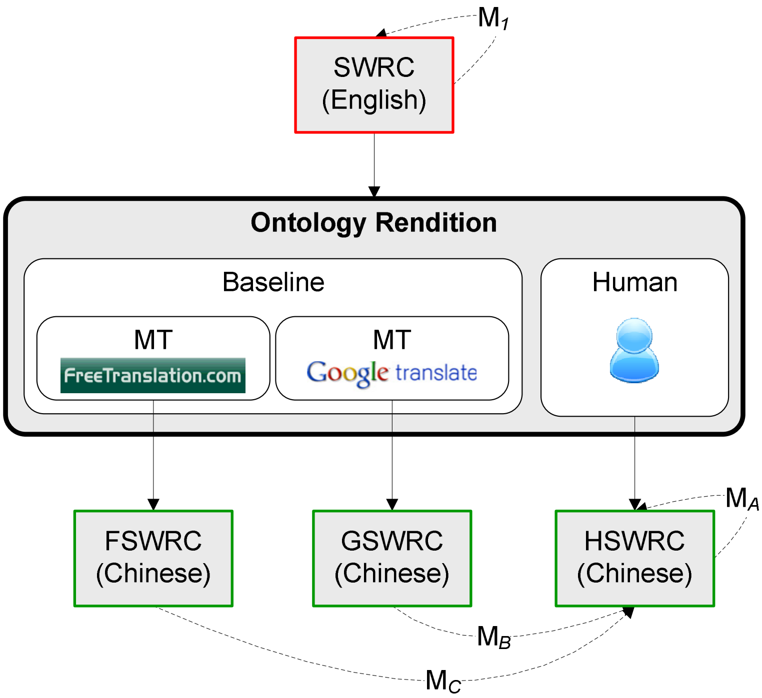


Figure 3-6. Experiment One Overview

Though it may be argued that the HSWRC ontology may bias the findings since it was created by the author of this thesis, however, as the goal of the experiment is to examine how different translations (i.e. different renditions of the same ontology) will impact on the mapping quality, the HSWRC ontology simply serves as one possible rendition of the SWRC ontology.

Figure 3-6 illustrates the experimental steps undertaken and the mappings are conducted as follows: (1) the SWRC ontology is mapped to itself using each algorithm from the Alignment API (recall there are a total of eight algorithms as discussed previously in chapter 2, section 2.5.2) to generate a gold standard as M_1, with matches in English. (2) The HSWRC ontology is then mapped to itself using the same algorithms to generate: M_A - containing matches in Chinese. M_A is then compared to M_1 manually (by the author of this thesis). If exactly the same pairs of matches are validated in M_A as those found in M_1, then M_A is essentially the Chinese gold standard for this experiment. (3) Next, the GSWRC ontology and the FSWRC ontology are each mapped to the HSWRC ontology to create the mappings M_B and M_C respectively (using the same eight algorithms from the Alignment API), both containing matches in Chinese. (4) Finally, M_B and M_C are evaluated against M_A. Note that M_1, M_A, M_B and

[55] http://protege.stanford.edu

M_C each contain eight sets of matches, as it is of interest to investigate how different matching algorithms are affected by the ontology rendition process.

One possible experimental outcome is that M_B and M_C show the same set of matches as M_A (assuming M_A is validated as a reliable gold standard in Chinese by the author of this thesis). This would mean that the translation of ontology labels did not have any side effects on the ontology rendition process. Since no matter who translated these labels, the renditions led to the same matching outcome when the rendered ontologies were mapped to one another. Another possible outcome is that M_B and M_C are shown to be of poor quality (i.e. low precision, recall and f-measure) when evaluated against M_A, it would mean that the translation of ontology labels has introduced noise into the ontology rendition process. This second outcome was later found to be the case by the experimental findings. This is presented and discussed next.

3.4.1.2. Findings and Analysis

Regardless of the matching algorithms used, the exact same sets of matches generated in M_I were found in M_A, where each resource matches to itself, i.e. a total of 128 matches was generated in M_A. It is thus with confidence that M_A can be considered as the gold standard in Chinese. Based on comparisons made to M_A, the precision, recall and f-measure of M_B and M_C are generated as shown in figure 3-7. The matches generated by the eight matching algorithms[56] in M_B and M_C are presented on the x-axis. The values on the y-axis range between 0.0 and 1.0. Precision scores are illustrated by blue bars, recall scores are illustrated by red bars and the f-measure scores are illustrated by green bars. For example, the *StringDistAlignment* algorithm (numbered 8 on the x-axis) generated 1.0 precision (blue bars), over 0.25 recall (red bars) and approximately 0.50 f-measure (green bars) in M_B and M_C. Note that in this evaluation, a match is considered correct as long as it is included in the gold standard regardless of its confidence level. Such an evaluation approach aims to measure the maximum precision, recall and f-measure scores that can be achieved in this experimental setting.

[56] In all the experiments presented in this chapter, the ClassStructAlignment algorithm is accompanied by the StringDistAlignment algorithm because it can only be executed with another algorithm from the API.

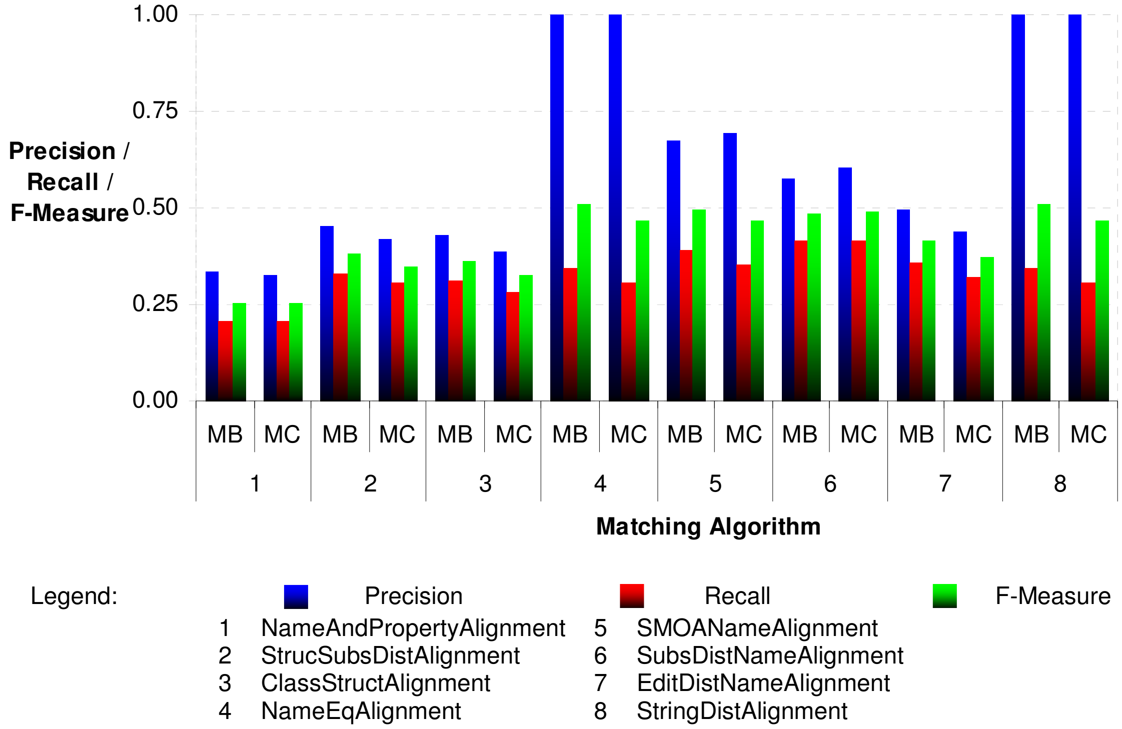


Figure 3-7. Experiment One Evaluation Results

It is clear from figure 3-7, that two string-based matching algorithms, namely *NameEqAlignment* and *StringDistAlignment* had the highest precision score of 1.00. However, no particular matching algorithm was able to generate remarkably high recall scores, including the aforementioned two algorithms. As a result, only less than 0.5 of the f-measure scores were achieved across eight matching algorithms. This means that only less than half of the matches were regenerated (post the ontology rendition process), which is rather low. A noticeable trend in figure 3-7 is that, generally, lexicon-based matching algorithms (i.e. *NameEqAlignment*, *SMOANameAlignment*, *SubsDistNameAlignment*, *EditDistNameAlignment* and *StringDistAlignment*) had higher precision, recall and hence f-measure scores compared to structure-based matching algorithms (i.e. *NameAndPropertyAlignment*, *ClassStructAlignment* and *StrucSubsDist-Alignment*). As structure-based techniques build upon the outcome of lexicon-based techniques, in the case of the latter performing poorly it interrupts the matching effectiveness of the former, as is shown in this experiment.

In both M_B and M_C, regardless of the matching algorithms used, the precision score is always higher than its corresponding recall score. This suggests that a considerable number of correct matches are found (and in some cases, 100% of the matches generated are correct, i.e. in the case of the *NameEqAlignment* and the *StringDistAlignment* algorithm), however, they are always an incomplete set compared to the gold standard. The average f-measure achieved in M$_B$ is 0.4272, and the average f-measure achieved in M$_C$ is 0.3992. This finding suggests that translations returned from the GoogleTranslate API are of a slightly higher quality (i.e. closer to the human translations) than those returned from the FreeTranslation online translator in this experiment. However, with all results having an f-measure of below 0.5, it is clear that translation noise has been introduced during the ontology rendition process.

The findings from experiment one show that the ontology label translation process has introduced noise for the subsequent matching step. Translation noise is evident in the matching outcome, since different renditions of the same ontology did not generate the same matching results when mapped to one another. Though one could argue that the ontology label translations in the HSWRC ontology may be biased (i.e. labels were translated in a way that they would not generate matches with the FSWRC ontology or the GSWRC ontology) since the author of this thesis was involved in its construction. However, as the goal of this experiment is to investigate the impact from the act of translation on the mapping outcome, the HSWRC ontology should be considered as one example of many other possible renditions. Also, since it was generated prior to any knowledge of the matching outcome, possible bias is minimised. To further investigate the impact of ontology label translations on the mapping process, a second experiment is conducted. This is discussed next.

3.4.2. Experiment Two

The goal of experiment two is to investigate how differing translations will affect mapping outcome when the same ontologies were mapped to each other before and after the ontology rendition process. The experimental setup is outlined in section 3.4.2.1, and the findings are presented in section 3.4.2.2.

3.4.2.1. Experimental Setup

An overview of the experimental setup is shown in figure 3-8. Two renditions of the SWRC ontology (in Chinese) and another two renditions of the ISWC ontology (also in Chinese) are generated through the ontology rendition process. The GSWRC and the GISWC ontology are created using the GoogleTranslate API. The FSWRC and the FISWC ontology are created using the FreeTranslation online translator.

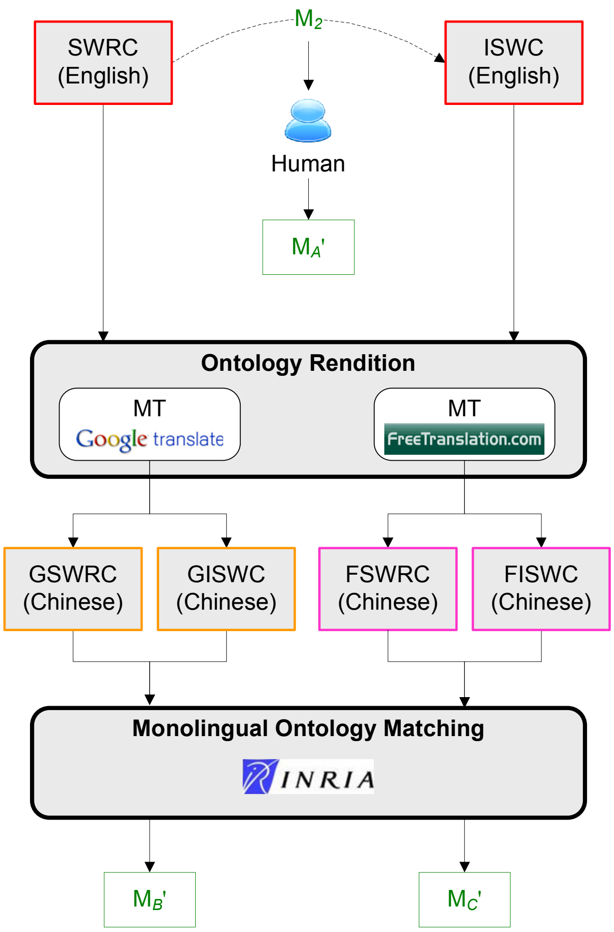


Figure 3-8. Experiment Two Overview

The mapping procedures carried out are as follows: (1) the original English SWRC ontology is mapped to the original English ISWC ontology to generate M_2 (in English). M_2 contains eight sets of matches that are generated using the eight different matching algorithms. (2) The GSWRC ontology is mapped to the GISWC ontology (again using eight matching algorithms) to generate M_B' (in Chinese). (3) Similarly, the FSWRC ontology is mapped to the FISWC ontology to generated M_C' (in Chinese). Note that each M_B' and M_C' contain eight set of matches (since eight different matching algorithms were applied). (4) To evaluate the quality of M_B' and M_C', they are compared against M_2. Since M_2 contains matched resources in English, the labels of these resources were translated manually to Chinese by the author of this thesis as M_A'. M_A' (in Chinese with a total of 57 matches) is then regarded as the reference standard for the evaluations of M_B' and M_C'. Note that M_2 is not a validated gold standard per se (it is generated by MOM techniques without verifications from human experts) in this experiment, it should be regarded as a reference for matches generated before ontology rendition. In other words, this experimental setup examines whether the MOM techniques is able to re-generate the same set of matches after the translations of ontology labels take place. Also note that although M_A' is created by this author, however, this process does not introduce bias into the reference standard. Because from M_2 to M_A', it is a simple case of establishing which pair of resources were matched to each other. When M_B' and M_C' are compared to M_A', whether the same pairs of matches were generated post ontology rendition is investigated.

3.4.2.2. Findings and Analysis

The same evaluation metrics are used in the second experiment as used in the first experiment, where a pair of matched resources is considered correct as long as it is found in the reference standard regardless of its confidence level. Each match set (in M_B' and M_C') that was generated using a specific matching algorithm is always evaluated against the gold standard that used the same matching algorithm (in M_A'). The evaluation results of M_B' and M_C' are shown in figure 3-9.

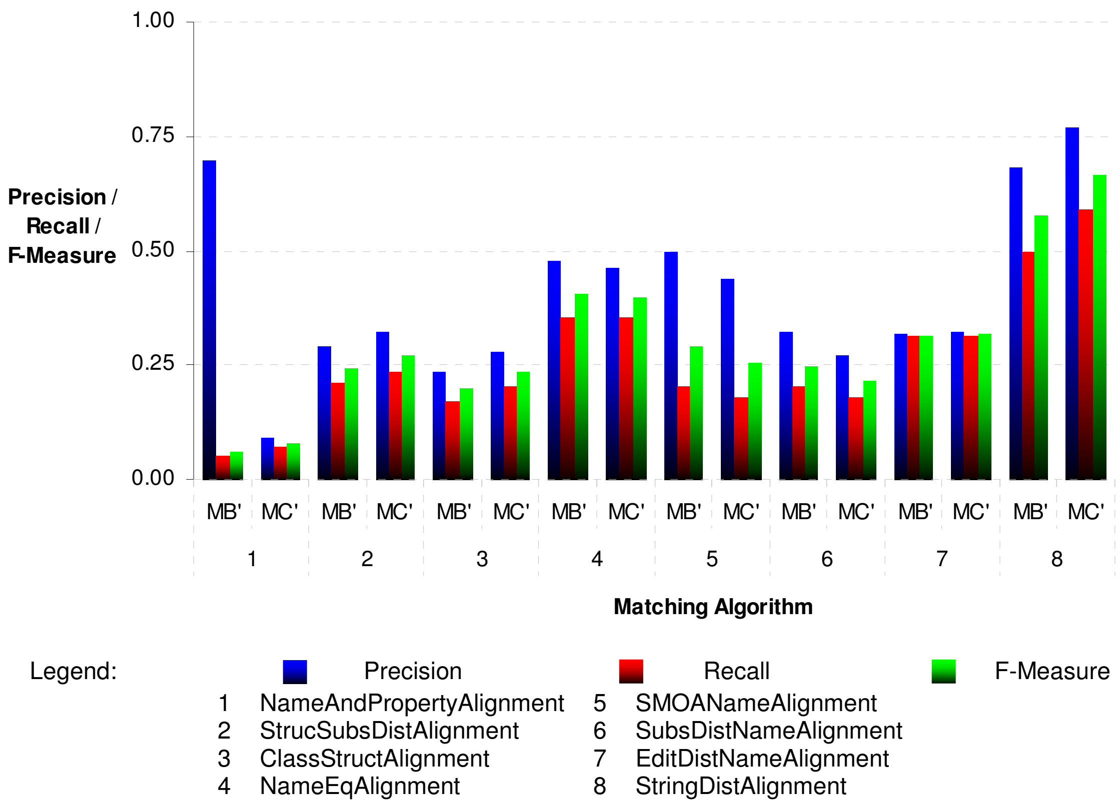


Figure 3-9. Experiment Two Evaluation Results

The *StringDistAlignment* algorithm had the highest precision and recall in this experiment, thus yielding the highest f-measure in both M_B' and M_C'. Similar to the findings from experiment one, lexicon-based matching algorithms generally had higher precision, recall and hence f-measure scores compared to structure-based matching algorithms. The mean f-measure in M_B' is 0.2927 and 0.3054 in M_C', which suggests that the FreeTranslation online translator had a slightly better performance than the

Google Translate API in this experiment. Nevertheless, the low f-measure scores found in this experiment indicate that the mappings generated are of rather poor quality. It is clear from experiment two that the MOM techniques were unable to simply re-generate the same set of matches between the same ontology pair post ontology rendition. These findings further confirm what was previously shown in experiment one: it is difficult for MOM algorithms to generate high quality matches when the ontology labels have been translated during the ontology rendition process. Conclusions drawn from the two experiments are discussed next.

3.4.3. Conclusions

It is shown through the experiments presented in this chapter that translation noise is introduced during the ontology rendition process, which have had a negative impact on the quality of the mappings subsequently generated using MOM techniques. Translation noise in the context of CLOM differs from the traditional sense. Traditionally, noise in the context of localisation can be understood as translations that do not meet the requirements of the target community. In the context of CLOM however, translation noise can be understood as translations that lead to incorrect matches or neglect correct matches (the scale of the translation noise problem is shown through the below 1.0 precision, recall and f-measure in both experiments). Also note that this difference means that reducing translation noise in CLOM is primarily concerned with selecting translations that will ensure the success of the subsequent MOM step. It does not concern selecting translations from a linguistic view point that is motivated by localisation. Translation noise (in the context of CLOM) exists as long as ontology labels are translated (for MOM techniques), it is not a result from the use of MT tools, it is in fact a result from the simple act of (ontology label) translation. It may be argued that since the experiments presented in this chapter only concern two ontologies, the conclusions drawn are not representative. It is thus important to note that the ontologies used in the experiments are not designed to be an exhaustive list, but rather examples of mapping scenarios. These example scenarios present this research with a ground for investigating translations that take place in the context of CLOM.

The author of this thesis manually examined the translations conducted in both experiments, and categorised three main types of translation noise. Table 3-1 gives an overview of the translation noise presented during the ontology rendition process in both experiments. The percentages shown in table 3-1 are calculated as: the sum of a particular type of translations divided by the sum of the labels to be translated. For example, in the case of the GSWRC ontology, the total number of inadequate translations presented is 19, the total of labels to be translated is 128, hence the percentage is 14.84% (i.e. 19/128).

Table 3-1. Translation Noise during Ontology Rendition

Renditions	**Noise**	**Translation Type**		
		Inadequate Translations	Synonymic Translations	Incorrect Translations
GSWRC	Count	19	26	15
	%	14.84%	20.31%	11.72%
FSWRC	Count	14	35	11
	%	10.94%	27.34%	8.59%
HSWRC	Count	0	29	0
	%	0	22.66%	0
GISWC	Count	12	16	5
	%	10.17%	13.56%	4.24%
FISWC	Count	16	16	2
	%	13.56%	13.56%	1.69%

Translation noise encountered in the experiments can be categorised as:

- **Inadequate translations.** These are translations that fail to adequately capture the concept of a resource label in a given CLOM context. These translations often grasp a general idea of the concept at hand, but fail to illustrate the exact terminology that is most suitable given the mapping tasks at hand. For example, as discussed in section 3.3.2, class labels *Conference* and *Meeting* were both translated to the same term that means "meeting" in Chinese. However, since conference is a specified type of meeting, the translation from MT was not adequate enough to capture the intended concept presented in the original ontology. This type of translations can lead to mismatches or prevent the generation of otherwise valid matches in the subsequent ontology matching step. In the aforementioned example, the source resource labelled as *Conference* is incorrectly mapped to the target resource labelled as *Meeting* in the target ontology as exact matches, whereas the source resource *Meeting* should have been matched instead. Such mismatches could be avoided if the MOM algorithms were presented with an adequate translation. One way to achieve this is to account the context of use for the labels in the source ontology.

- **Synonymic translations.** These are translations that capture the intended meaning of the given resource labels in the source ontology, but differ lexically from the labels presented in the target ontology. Such translations may not be considered as semantic issues per se, however they do present challenges for MOM techniques that rely on lexicon comparisons in the process of generating matches. For instance, synonymic translations can cause the absence of correct matches in the subsequent monolingual ontology matching step. A simple example can be: assuming there is a source and a target resource both illustrating the concept of rain. The target resource is labelled as *shower*, whereas the translation for the source resource is *rain*. An absence of a match between the two can occur given lexicon-based MOM techniques as the two labels have little in common with one another. Such cases can be improved if the ontology rendition process accounts for the semantics (i.e. the labels already used by the target ontology) in the target ontology, e.g. select translations that are lexically similar to what are presented in the target ontology in order to conclude good matches.

- **Incorrect translations.** These are translations which do not reflect the meaning of the given concept in any way. Incorrect translations can lead to mismatches or the absences of correct matches in situations similar to the scenarios presented above. Such situations can be improved or avoided if a pool of candidate translations are available and the contexts of use (i.e. other labels that surround the to-be-translated label) are known to the ontology rendition process.

To improve the quality of CLOM results generated using the baseline approach, it is clear from the experimentation that ontology label translations need to take the mapping context into account. An improved CLOM approach needs to recognise which translations are appropriate in the given mapping scenario. A translation is appropriate if it successfully leads the MOM techniques to generate the correct mapping (given such a correct mapping exists). As the goal of CLOM is to generate quality mappings between ontologies that are labelled in different natural languages, the translations of the ontology labels merely serve as a means to an end whereby it should supply the MOM techniques with rendered ontologies that are likely to lead to good matches. In other words, the translations of ontology labels should be purposely conducted depending on the specific CLOM scenario. To achieve this, ontology labels should not be translated in isolation from the ontologies involved in a given CLOM scenario, the translations should centre on the semantics that are already embedded in these ontologies. Hence arising from the results of the experimentation conducted and described in this chapter, the novel concept of appropriate ontology label translation (AOLT) for CLOM is defined by the author. An appropriate ontology label translation

(AOLT) in the context of cross-lingual ontology mapping is defined as one that is most likely to maximize the success of the subsequent monolingual ontology matching step.

This idea should not be confused with generating translations with the purpose of localisation, for example, in the context of ontology localisation (discussed in chapter 2, section 2.4.2). The goal of ontology localisation is to generate ontologies that are adapted "to a particular language and culture" [Suárez-Figueroa & Gómez-Pérez, 2008], whereby the translations of ontology labels is a form of natural language processing. In contrast, the AOLT concept is concerned with searching for appropriate translations (from a mapping point of view) that are believed to be the ones most likely to enhance the matching ability of the subsequent MOM step, but not necessarily the most linguistically correct translations (from a localisation point of view).

3.5. Summary

In this chapter, an implementation of the baseline approach to CLOM is presented. The effectiveness of the baseline approach system in CLOM scenarios, particularly the ontology rendition process, is examined in two experiments. It is shown with evidence that translation noise can have a negative impact on the subsequent MOM step in the baseline system. Based on the conclusions drawn from these experiments, the concept of AOLT is proposed. Methods to achieve AOLT for the purpose of improving CLOM quality, as well as the evaluations of their effectiveness form the basis for the rest of the work presented in this thesis.

4 PROTOTYPE ONE: SOCOM

4.1. Chapter Overview

This chapter discusses the AOLT concept in detail and presents an initial CLOM prototype: SOCOM to realise the proposed concept. The design, implementation and evaluation of SOCOM are also presented. In addition, a case study of SOCOM in a cross-lingual adaptive retrieval and composition system is presented in this chapter. This chapter is organised as follows. Section 4.2 discusses the concept of AOLT. Section 4.3 presents the design of the SOCOM system that integrates an AOLT process to achieve cross-lingual ontology mapping. Section 4.4 discusses the implementation of SOCOM. Section 4.5 presents the evaluation of SOCOM in two CLOM experiments involving ontologies of the research and bibliography domain in Chinese, English and French. Section 4.6 demonstrates SOCOM in a case study whereby cross-lingual information retrieval (CLIR) is achieved through the use of cross-lingual ontology mapping. Finally, section 4.7 concludes the chapter with a summary.

The initial proposal of the SOCOM system (discussed in section 4.3) has been published in the paper titled *Multilingual Ontology Mapping: Challenges and a Proposed Framework* at the Symposium on Matching and Meaning (AISB 2009 Convention), ISBN 1902956842, pp. 32-35, in April 2009. The evaluation approach of the SOCOM system (discussed in section 4.5) has been published in a poster titled *Evaluation of a Semantic-Oriented Approach to Cross-Lingual Ontology Mapping* at Knowledge Engineering and Knowledge Management by the Masses (EKAW 2010), CEUR-WS Vol. 674, in October 2010. Findings from experiment one (discussed in section 4.5.1) and the case study (discussed in section 4.6) have been published in a paper titled *Cross-Lingual Ontology Mapping and Its Use on the Multilingual Semantic*

Web at the 1st Workshop on the Multilingual Semantic Web collocated at the 19th World Wide Web conference (WWW 2010), CEUR-WS Vol. 571, pp. 13-20 in April 2010.

4.2. The Concept of Appropriate Ontology Label Translation

The concept of appropriate ontology label translation (AOLT) was first introduced in chapter 3 (section 3.4.3). The basis of the AOLT concept is that it is useful to differentiate between translations that take place in the context of ontology localisation and translations that occur in the context of cross-lingual ontology mapping. In ontology localisation, the translation of labels aims to adapt the ontology to a particular language and culture. In cross-lingual ontology mapping, the translation of labels aims to adapt to the needs of the subsequent monolingual matching process in an effort to generate high quality cross-lingual ontology mapping results. As shown in the experiments presented in chapter 3 (section 3.4), translation noise (i.e. translations that neglect correct mappings or lead to incorrect mappings) can be introduced during ontology rendition which subsequently lead to poor matching quality. To improve this situation, the concept of AOLT can be applied. *An appropriate ontology label translation (AOLT) in the context of cross-lingual ontology mapping is one that is most likely to maximize the success of the subsequent monolingual ontology matching step.*

The core idea of the AOLT concept is: translations that take place in the context of CLOM should be mapping-oriented as these translations should facilitate MOM techniques in the generation of quality mappings. There can be various ways to realise the AOLT concept. For instance, human CLOM experts specialising in certain domains and familiar with specific natural language pairs can manually select AOLT results in a given CLOM scenario. Another example to achieve AOLT results can be rule-based, e.g. CLOM results can be aggregated over time to help determining which translations are appropriate in the given domain and specific natural language pair. This effectively creates a translation memory[57] specifically for CLOM scenarios involving specific domains and natural languages, which can be used for future translations carried out in the same CLOM context. This thesis however, focuses on realising the AOLT concept without the involvement of a user or translation memories that are likely to require the maintenance of a user. In other words, this thesis aims to select AOLT results based on

[57] A translation memory is "an archive of existing translations, structured in such a way as to promote translation re-use" [Macklovitch et al., 2000].

information drawn from the ontologies involved in a CLOM scenario. To better understand the AOLT concept, an example is shown in figure 4-1, where the source ontology is labelled in Chinese and is mapped to an English target ontology. To achieve cross-lingual ontology mapping, the AOLT process is performed to first translate the labels in the Chinese ontology into English. The source class 摘要 (meaning "abstract" or "summary") has candidate translations *abstract* and *summary*. To determine the most appropriate translation (underlined in figure 4-1), consider three scenarios.

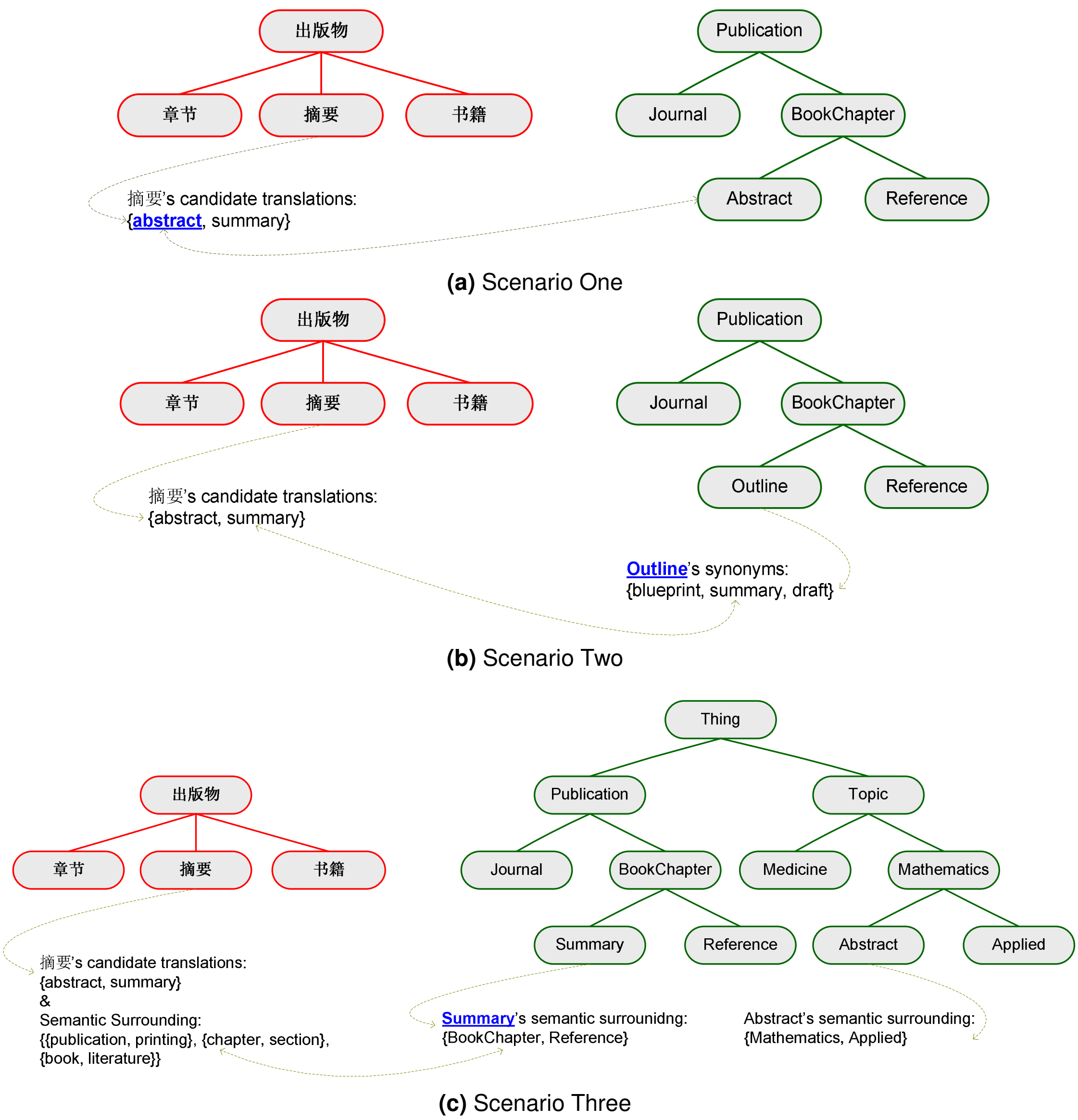


Figure 4-1. Examples of AOLT in the Context of CLOM

Figure 4-1-a demonstrates a situation where a class labelled as *Abstract* exists in the target ontology. In this case, *Abstract* would be a more appropriate translation than *summary*, since it is more likely for MOM techniques to generate a match.

Figure 4-1-b illustrates another scenario where the target ontology contains a class labelled *Outline*. From a thesaurus or dictionary, it can be determined that *Outline* is a synonym of the candidate translation *summary*, therefore, instead of using either *abstract* or *summary*, *Outline* is chosen as the appropriate translation since it is the exact label used by the target ontology.

Figure 4-1-c shows a third scenario where both *Abstract* and *Summary* exist in the target ontology, the appropriate translation is then concluded by analysing the semantic surroundings. In this thesis, the semantic surrounding of an entity refers to the labels that are used by the immediate surrounding nodes of this entity. For a class entity *C*, its surrounding nodes include its immediate associated node(s) that is one level higher and/or lower to *C* in the given ontological hierarchy. For a property entity *P* (either datatype or object), its surrounding is defined as the entity or entities which *P* restricts. For an individual (or instance) entity *I*, its surrounding node is defined as the class entity or entities which *I* belongs to. It is recognised that the semantic surrounding of an entity can include a broader range of nodes than just the immediate associates. At the broadest extreme for example, all the semantics that are contained in the given ontology can be considered as the semantic surrounding of a node. However, as the range increases, the overlap of semantic surroundings between entity E_1 and entity E_2 increases. This increased overlap will narrow the distinctions between the semantic surroundings among entities. In this thesis, in order to maintain a distinctive representation for a given entity from another entity in the same ontology, the semantic surroundings thus only concern the immediate associated nodes. In the third scenario, the source class 摘要 has a super-class 出版物 (with candidate translations *publication* and *printing*), two sibling-classes 章节 (with candidate translations *chapter* and *section*) and 书籍 (with candidate translations *book* and *literature*). Its semantic surrounding therefore include: {*publication*, *printing*, *chapter*, *section*, *book*, *literature*}. Similarly, in the target ontology, the semantic surrounding of the class *Abstract* can be collected as: {*Mathematics*, *Applied*}, and the semantic surrounding of the class *Summary* would include: {*BookChapter*, *Reference*}. Using string comparison techniques such as edit distances, it can be determined that the strings in the surrounding of the target class *Summary* are more similar to those of the source class. *Summary* therefore would be the appropriate translation in this case.

Given a pair of ontologies in a mapping scenario, one immediate improvement on the selection of the translations is to take into account the semantics embedded in both ontologies during the label translation process. Given ontologies O_1 and O_2 in a CLOM scenario, the minimum semantics that can be taken into account by the AOLT process is the data already coded in these ontologies. In other words, a basic AOLT process focuses on what is always available in any CLOM scenario, i.e. a source and a target ontology and the semantics within them to influence the selection of appropriate translations. For example, the semantic surrounding of an O_1 entity illustrates its context-of-use in the source ontology, and the entity labels in O_2 present the AOLT process with selection criteria, as the example shown in figure 4-1. This minimum intake of ontology semantics can be considered as a basic modelling of the AOLT process. Prototype one: SOCOM aims to investigate whether such a basic AOLT process can improve the CLOM quality, through the implementation (section 4.4) and the evaluation (section 4.5) of this prototype. The design of SOCOM is discussed next.

4.3. SOCOM Design

An initial Java-based prototype of the Semantic-Oriented Cross-lingual Ontology Mapping (SOCOM) system is designed to incorporate the basic model of the AOLT process (discussed in previous section) to achieve CLOM. This section presents the design of the SOCOM system.

SOCOM allows a user to generate mapping results between ontologies that are labelled in different natural languages. The flowchart in figure 4-2 illustrates the workflow of SOCOM. To achieve cross-lingual ontology mapping, SOCOM carries out seven steps including ontology parsing, label translation, synonym generation, AOLT selection, ontology rendition, MOM and match reconstruction, as follows.

- The *ontology parsing* step is responsible for extracting the labels and the semantic surroundings from a given ontology, which is performed on both the source and the target ontology.
- The *label translation* step is responsible for generating the candidate translations for the labels in the source ontology.
- The *synonym generation* step is responsible for generating the synonyms for the labels in the target ontology.

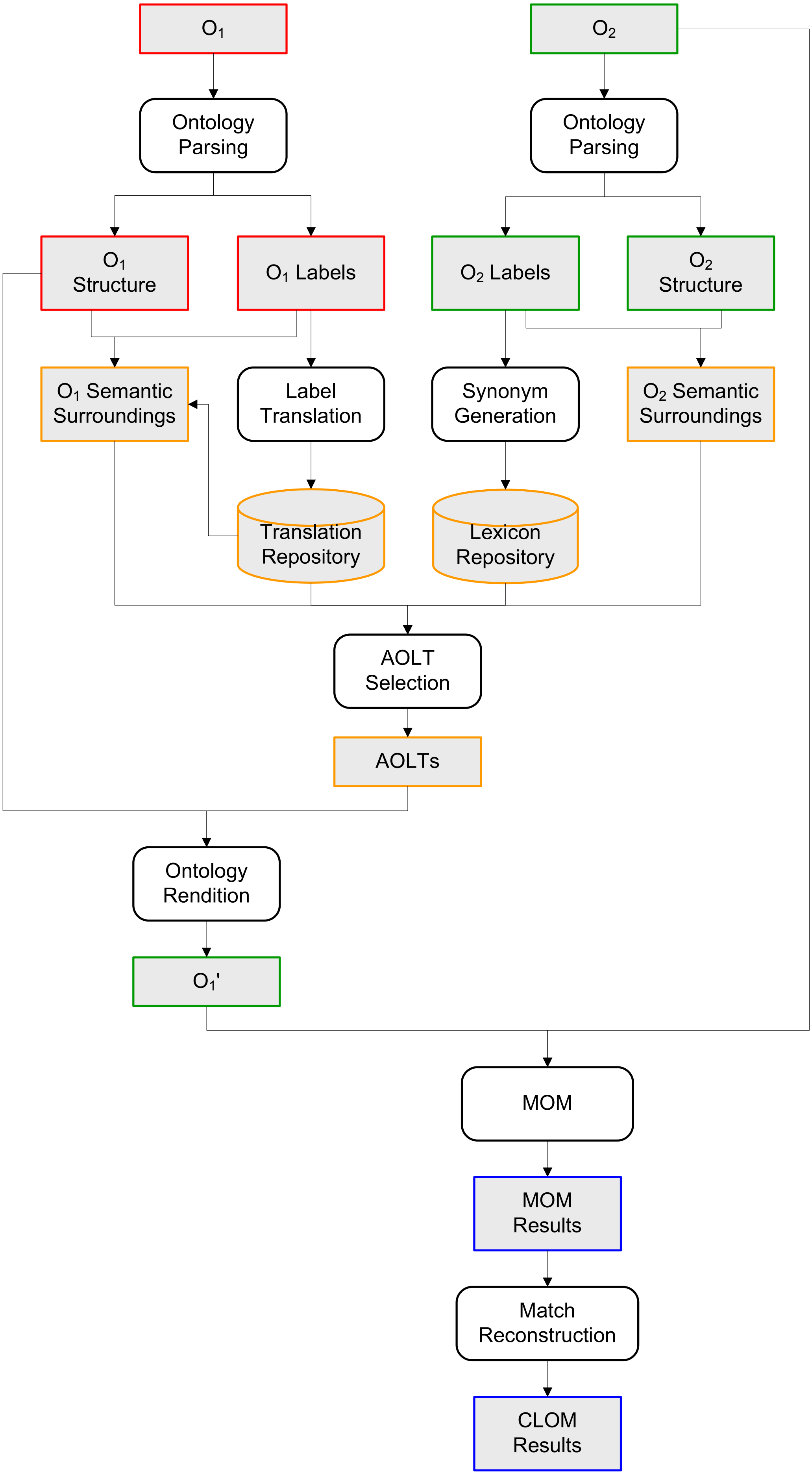


Figure 4-2. Workflow in SOCOM

- The *AOLT selection* step is responsible for generating AOLT results from the available translations and synonyms based on comparisons made between the source and the target semantic surroundings. Translation collisions are also resolved during this process before an AOLT result is stored.
- The *ontology rendition* step is responsible for generating a version of the source ontology that is labelled in that target natural language, using the AOLT results concluded during the AOLT selection step.
- The *MOM* step is responsible for generating matches between the rendered ontology (i.e. a converted source ontology with labels in the target natural language) and the target ontology.
- Finally, the *match reconstruction* step is responsible for generating matches between the source ontology and the target ontology, based on matches generated between the rendered ontology and the target ontology, and the AOLT results selected for source ontology labels.

The MOM step and the match reconstruction step in SOCOM are the same as in the baseline system described in chapter 3 (section 3.3.2). The innovative difference between SOCOM and the baseline system is the rendition of O_1'. More specifically, the difference lies in the translations of the O_1 labels during the rendition process. In the baseline system, translations of O_1 labels are achieved by MT tools independently of the mapping scenario. In other words, the baseline system ignores the ontologies that are involved in a mapping scenario, and conducts the label translations in isolation. In contrast, SOCOM aims to achieve appropriate translations for O_1 labels, whereby the translations are motivated by supporting the subsequent MOM process. The translation of ontology labels in SOCOM is not conducted in isolation of the ontologies involved in a mapping scenario. The semantics from the target ontology (i.e. the labels used by target entities and their semantic surroundings) are used to influence the translation outcome of the labels in the source ontology.

Note that the translations resulting from the AOLT process related to the same ontology: O_1 will differ depending on the given target ontology: O_2 in a mapping scenario, since the semantic data in O_2 will influence the selections of the AOLT results for O_1 labels. The rendered ontology O_1' should not be considered as a localised O_1, but simply an intermediate step in the cross-lingual ontology mapping process. More details on achieving AOLT results in SOCOM are discussed next.

Given ontologies O_1 and O_2 that are labelled in different natural languages, the semantics (i.e. labels and semantic surroundings) from both ontologies are extracted through the *ontology parsing* step (see figure 4-2). The resource labels from O_1 are sent to the *label translation* step to generate candidate translations in the natural language used by O_2, and later stored in the *translation repository*. The synonyms of the resource labels in O_2 are generated through the *synonym generation* step, and later stored in the *lexicon repository*. Knowing the location of a node in a given ontology, the semantic surrounding of this node can be collected as discussed in section 4.2. The output from this process is shown as *O_1 semantic surroundings* and *O_2 semantic surroundings* in figure 4-2, which include the semantic surroundings for all the classes, properties and individuals in an ontology. Note that the surroundings of resources in O_1 are labelled in the original source natural language. In order to compare the semantic surroundings of a source entity to that of a target entity, these semantic surroundings need to be labelled in the target natural language. Hence, in figure 4-2, the *O_1 semantic surroundings* are generated with three inputs including: the *O_1 labels*, the *O_1 structure* and the *translation repository*. This differs from the generation of the *O_2 semantic surroundings*, which requires two inputs only: the *O_2 labels* and the *O_2 structure*.

Figure 4-3 illustrates how a candidate AOLT (i.e. before verifying this AOLT result is collision-free) is selected for a label L_1 in the source ontology O_1. For each candidate translation of L_1 (stored in the translation repository in figure 4-2), it is compared to the target labels and their synonyms (stored in the lexicon repository in figure 4-2) using string comparison techniques. Three possible outcomes (shown in figure 4-3, the decision point of *candidate translation = O_2 Label/Synonym* has two possible outcomes: *yes* or *no*, with *yes* further refined to either *one-to-many* or *one-to-one* match) can derive from this comparison process as follows:

- If a one-to-one match (i.e. the candidate translation is the same with a target resource label, or a synonym of a target resource label) is found, the target label or the matched synonym's corresponding target label is selected as the candidate AOLT. A match in this context refers to two character strings with edit distance of zero. This string comparison technique is further explained in section 4.4.
- If one-to-many matches (i.e. multiple target labels and/or synonyms in the lexicon repository are the same with a given candidate translation) are found, the semantic surroundings of the corresponding target labels are

collected and compared to the semantic surrounding of the source label in question. The target label with semantic surroundings that are most similar (i.e. with lowest aggregated edit distance) to those of the source resource is chosen as the AOLT.

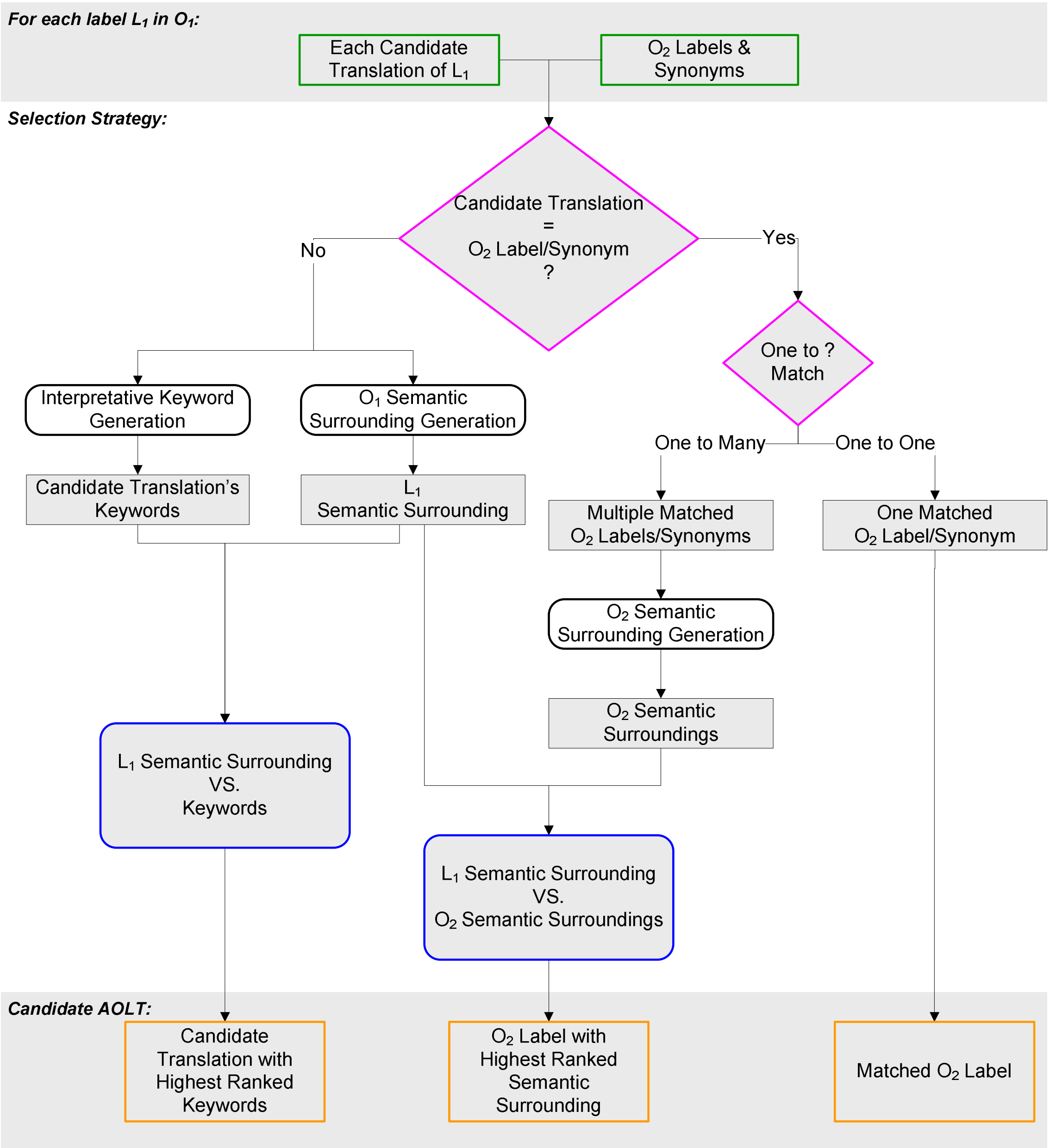


Figure 4-3. The AOLT Process in SOCOM

- If no match is found in the lexicon repository, for each candidate translation, a set of interpretative keywords are generated to illustrate the meaning of this candidate. The candidate with keywords that are most similar (i.e. having lowest aggregated edit distance score) to the source label's semantic surrounding is deemed as the AOLT result. Interpretative keywords can be generated from resources such as dictionaries and encyclopaedias. In SOCOM, interpretative keywords for candidate translations are generated using Wikipedia. Implementation details of the keyword generation process are discussed in section 4.4.

After a candidate AOLT result is generated for a label in O_1, it needs to be collision-free (i.e. no two or more labels have the same AOLT result) to be considered as the actual AOLT result. Collisions can occur during the AOLT selection process and need to be resolved in order to generate O_1'. For example, the candidate AOLT for several labels in O_1 may lead to the same target label, or the same synonym of a target label, or different synonyms but all corresponding to the same target label. These situations are effectively a result of many-to-one matches (not included in the bullets above, as the above cases concern the possible outcomes regarding one and only one candidate translation at a time), and are referred to as translation collisions in this thesis. Such collisions must be resolved in order to generate well-formed URIs in O_1'. How translation collisions are resolved and the technologies used to realise the AOLT selection process described thus far are presented next.

4.4. SOCOM Implementation

This section discusses the techniques and technologies used in the implementation of the first prototype: SOCOM to achieve the processes shown in figure 4-2. The complete Java code of SOCOM can be found at `root/SOCOM/src/`, and the Jar files required to run SOCOM can be found at `root/SOCOM/bin/` on the DVD.

Ontology parsing: SOCOM uses the Jena Framework version 2.5.5 to parse the formally defined ontologies. The Jena Framework was chosen because it is open source and supports the reading and writing of both RDF and OWL ontologies (which is the focus of this thesis). A code snippet using the Jena Framework to load a locally stored ontology, iterate through the classes from within and extract the class labels is presented in appendix C, section C.3, figure C-5. The Jena Framework is also used to generate the semantic surroundings of a given ontological entity. Figure C-6 in appendix C, section C.3 presents a code snippet of the generation of semantic surrounding for an ontology class.

Translating O_1 labels: to collect candidate translations for ontology labels in O_1, the GoogleTranslate API version 0.5 and the WindowsLive[58] translator are used. These MT tools were chosen for SOCOM as they represent the state of the art in statistical MT (discussed in chapter 2, section 2.6). In the same way as the baseline system (discussed in chapter 3, section 3.3.2), concatenated ontology labels are split to their constituent words in natural language format before passed onto the integrated MT tools (as illustrated in appendix C, figure C-2 and figure C-3). Figure C-7 in appendix C, section C.3 presents a code snippet of how the candidate translations are achieved.

Generating synonyms for O_2 labels: English synonyms of the ontology labels in O_2 (required for experiment one discussed later in section 4.5.1) are generated by querying the WordNet[59] thesaurus, version 2.0 via the RiTaWN[60] API as well as the Dictionary.com API[61]. These thesauri are chosen since there are readily available Java APIs that offer easy access to their content. French synonyms of the ontology labels in O_2 (required for experiment two discussed later in section 4.5.2) are generated by querying synonyms-fr.com[62]. The synonyms returned from this website are parsed using the HTML Parser[63] version 2.0. The synonyms-fr.com was chosen because it provides French synonyms in a nested fashion that can be parsed by readily available Java libraries such as the HTML Parser. Figure C-8 in appendix C, section C.3 presents a code snippet of the generation of synonyms for a target individual label.

Generating the translation repository & the lexicon repository: as mentioned in the previous section, the translation repository contains the labels used in the source ontology and their corresponding candidate translations, and the lexicon repository contains the target labels and their respective synonyms. Both repositories are formatted in XML and stored in the eXist DB[64] version 1.0rc. The eXist database was chosen because it is open source, it supports XML data management and features efficient XQuery[65] and XPath[66] processing. Figure 4-4 gives an example of the translation repository generated for a source ontology labelled in Chinese. The Document Type

[58] http://www.windowslivetranslator.com/Default.aspx
[59] http://wordnet.princeton.edu
[60] http://www.rednoise.org/rita
[61] http://developer.dictionary.com/products
[62] http://www.synonyms-fr.com/
[63] http://htmlparser.sourceforge.net/
[64] http://exist.sourceforge.net
[65] XQuery is a functional programming language that is designed to query collections of XML data. More on XQuery can be found at http://www.w3schools.com/xquery/default.asp
[66] XPath is a query language for navigating through elements and attributes in an XML document. More on XPath can be found at http://www.w3schools.com/xpath/

Definition (DTD) declared for the translation repository can be found in appendix D, section D.2, figure D-1. Figure 4-5 presents an example of the lexicon repository generated for a target ontology labelled in English. The DTD used by the lexicon repository can be found in appendix D, section D.2, figure D-2.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TranslationRepository SYSTEM "TranslationRepository.dtd">
<TranslationRepository>
   …
    <Result>
        <SourceID>SC1</SourceID>
        <SourceValue>院所</SourceValue>
        <CandidateCollection>
            <Candidate>
                <CandidateID>STC1-SC1</CandidateID>
                <CandidateValue>Institutions</CandidateValue>
            </Candidate>
            <Candidate>
                <CandidateID>STC2-SC1</CandidateID>
                <CandidateValue>Institutes</CandidateValue>
            </Candidate>
        </CandidateCollection>
    </Result>
    …
    <Result>
        <SourceID>SC14</SourceID>
        <SourceValue>管理人员</SourceValue>
        <CandidateCollection>
            <Candidate>
                <CandidateID>STC27-SC14</CandidateID>
                <CandidateValue>Managers</CandidateValue>
            </Candidate>
            <Candidate>
                <CandidateID>STC28-SC14</CandidateID>
                <CandidateValue>Management staff</CandidateValue>
            </Candidate>
        </CandidateCollection>
    </Result>
    …
</TranslationRepository>
```

Figure 4-4. An Example of the Translation Repository

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LexiconRepository SYSTEM "LexiconRepository.dtd">
<LexiconRepository>
    …
    <Result>
        <TargetID>TC3</TargetID>
        <TargetValue>Student</TargetValue>
        <SynonymCollection>
            <Synonym>
                <SynonymID>TSN21-TC3</SynonymID>
                <SynonymValue>apprentice</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN22-TC3</SynonymID>
                <SynonymValue>auditor</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN23-TC3</SynonymID>
                <SynonymValue>junior</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN24-TC3</SynonymID>
                <SynonymValue>learner</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN25-TC3</SynonymID>
                <SynonymValue>decoder</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN26-TC3</SynonymID>
                <SynonymValue>observer</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN27-TC3</SynonymID>
                <SynonymValue>graduate</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN28-TC3</SynonymID>
                <SynonymValue>novice</SynonymValue>
```

```
            </Synonym>
            <Synonym>
                <SynonymID>TSN29-TC3</SynonymID>
                <SynonymValue>pupil</SynonymValue>
            </Synonym>
            <Synonym>
                <SynonymID>TSN30-TC3</SynonymID>
                <SynonymValue>valedictorian</SynonymValue>
            </Synonym>
        </SynonymCollection>
    </Result>
    …
</LexiconRepository>
```

Figure 4-5. An Example of the Lexicon Repository

The AOLT selection process invokes the repositories in the eXist database via the XML:DB[67] 1.0 API, to compare each candidate translation of a given source label to what is stored in the lexicon repository. This process is discussed earlier in section 4.3. If a one-to-one match (note that the match found in the lexicon repository can be either a target label used in O_2, or a synonym of a target label that is used in O_2) is found, the (matched target label or the matched synonym's corresponding) target label is selected as the AOLT. If one-to-many matches (i.e. when several target labels and/or synonyms in the lexicon repository are matched) are found, the semantic surroundings of the matched target labels are collected and compared to the semantic surroundings of the source label in question. If no match is found in the lexicon repository, for each candidate translation, a set of interpretative keywords are generated to illustrate the meaning of this candidate. This is achieved by querying Wikipedia[68] via the Yahoo Term Extraction Tool[69]. The code snippet in appendix C, section C.3, figure C-9 illustrates this process. An example output from term extraction is shown in figure 4-6, where key words are extracted for *conference* based on its definition from Wikipedia.

```
…
<Result>verbal interaction</Result>
<Result>gatherings</Result>
<Result>exhibition</Result>
<Result>presentation seminar</Result>
<Result>demonstration</Result>
…
```

Figure 4-6. An Example Output from Term Extraction

String comparison technique: edit distance is often used to compare the similarity between strings. "Given two character strings S_1 and S_2, the edit distance between them is the minimum number of edit operations required to transform S_1 into S_2" [Manning et al., 2008, p. 58]. Edit operations include insertion, deletion or replacement of a character in the given string. More details of edit distance can be found in [Manning et al., 2008]. In SOCOM, a space/case-insensitive edit distance

[67] http://xmldb-org.sourceforge.net/index.html
[68] http://www.wikipedia.org
[69] http://developer.yahoo.com/search/content/V1/ termExtraction.html

string comparison algorithm based on Nerbonne et al.'s method [Nerbonne et al., 1999] is used to compare labels (e.g. comparing a candidate translation to a target label) and collections of labels (e.g. comparing a set of interpretative keywords for a candidate translation to its corresponding source label's semantic surrounding) via the LingPipe API[70] version 3.8.0. This comparison algorithm used between two character strings is demonstrated in a code snippet shown in appendix C, section C.3, figure C-10. The comparison algorithm implemented for semantic surroundings (i.e. comparisons made between two collections of character strings) is demonstrated in a code snippet shown in appendix C, section C.3, figure C-11.

Translation collisions are resolved upon the conclusion of a final AOLT result. A summary of the resolution strategies used in SOCOM is shown in table 4-1. Translation collisions can occur between two or more source entity labels, however, the system only needs to be concerned with two entities at a time as collisions need to be solved immediately upon detection. For a given source label, if its AOLT is determined based on a match made to a target label or the synonym of a target label in the lexicon repository, the origin of this AOLT is categorised as *derived from target ontology*. In all other cases, the origins of the AOLT results are categorised as *derived without target ontology*. When a translation collision is detected between a pair of source entities E_1 and E_2, the origins of their AOLT results are verified. The entity with the AOLT that was derived from the target ontology keeps the collided term as its AOLT, and the other entity will search for an alternative translation as its AOLT, as shown in table 4-1, scenario i and iv. If both entities used the same strategy to determine their AOLT result, the latter entity will seek alternative translation as shown in table 4-1, scenario ii and iii.

To seek alternative translation, if the initial AOLT was derived with the help of the target ontology (i.e. a match made to a target label, or a synonym of a target label), the system searches among available synonyms (of a target label) until one is found that does not cause further collisions. If for the entity that is seeking an alternative translation, its AOLT result was derived without the help of the target ontology (i.e. based on keyword comparison made to the source resource surrounding), the system searches among the available keywords generated (for the candidate translation) until one is found that does not cause further collisions. However, it is possible that an alternative translation no longer exists when all the available candidate translations are deemed to be unsuitable (i.e. they cause further collisions). In such situations, the

[70] http://alias-i.com/lingpipe

system employs the same techniques as described in chapter 3 (section 3.3.2), whereby a unique integer is added to the end of the collided AOLT for the entity with no suitable alternatives. The code snippet shown in appendix C, section C.3, figure C-12 illustrates the main steps involved in the collision resolution process.

Table 4-1. Collision Resolution in SOCOM

Collision Scenario	AOLT Origin		Solution
	E_1	E_2	
i	derived with help from target ontology	derived without help from target ontology	E_1 keeps collided AOLT; E_2 seeks alternative
ii	derived with help from target ontology	derived with help from target ontology	
iii	derived without help from target ontology	derived without help from target ontology	
iv	derived without help from target ontology	derived with help from target ontology	E_2 keeps collided AOLT; E_1 seeks alternative

Generating O_1' and CLOM results: once AOLT results are identified for each resource label in O_1, O_1' is generated using the Jena Framework based on the original source ontology structure, as discussed previously in chapter 3 (section 3.3.2, also demonstrated by the code snippets shown in appendix C, section C.1, figure C-1, figure C-2 and figure C-3). Finally, O_1' is matched to O_2 to generate candidate MOM matches via the Alignment API version 3.6. The CLOM results are finally generated based on the MOM results and the translations for labels in O_1. This match reconstruction process is the same with the baseline system.

Summary: the implementation discussed thus far in this section is illustrated by the class diagram shown in figure 4-7. The class *SourceAnalysis* is responsible for extracting the labels from a given source ontology, populating and storing their corresponding candidate translations by calling the *TranslationService* class which breaks up concatenated labels via the *LabelReconstruction* class, and generates the semantic surrounding for a given source entity upon requests from the *AOLTResult-Selection* class when translation collisions are detected. The *TargetAnalysis* class is responsible for extracting the labels in the given target ontology, generating and storing their corresponding synonyms by calling the *LexiconService* class which splits concatenated labels into natural language formats, as well as generating the semantic surroundings upon requests from the *AOLTSelection* class when solving translation collisions. The AOLT results are then selected by the *AOLTSelection* class, and translation collisions are solved before the storing of these AOLT results. The *AOLTSelection* class also initiates the *KeywordGeneration* class when collisions must be solved by using interpretive keywords, and the *RankingService* class when semantic surrounding similarities need to be calculated. The *CasePuncuationDistance* class is responsible for comparing string similarities by invoking the *WeightedEditDistance* class. O_1' is generated by the *OntologyRendition* class which also concatenates the labels by calling the *LabelReconstruction* class. Finally, the *MatchingService* generates matches using various MOM algorithms and reconstructing these MOM matches based on the known AOLT results to create the final CLOM results.

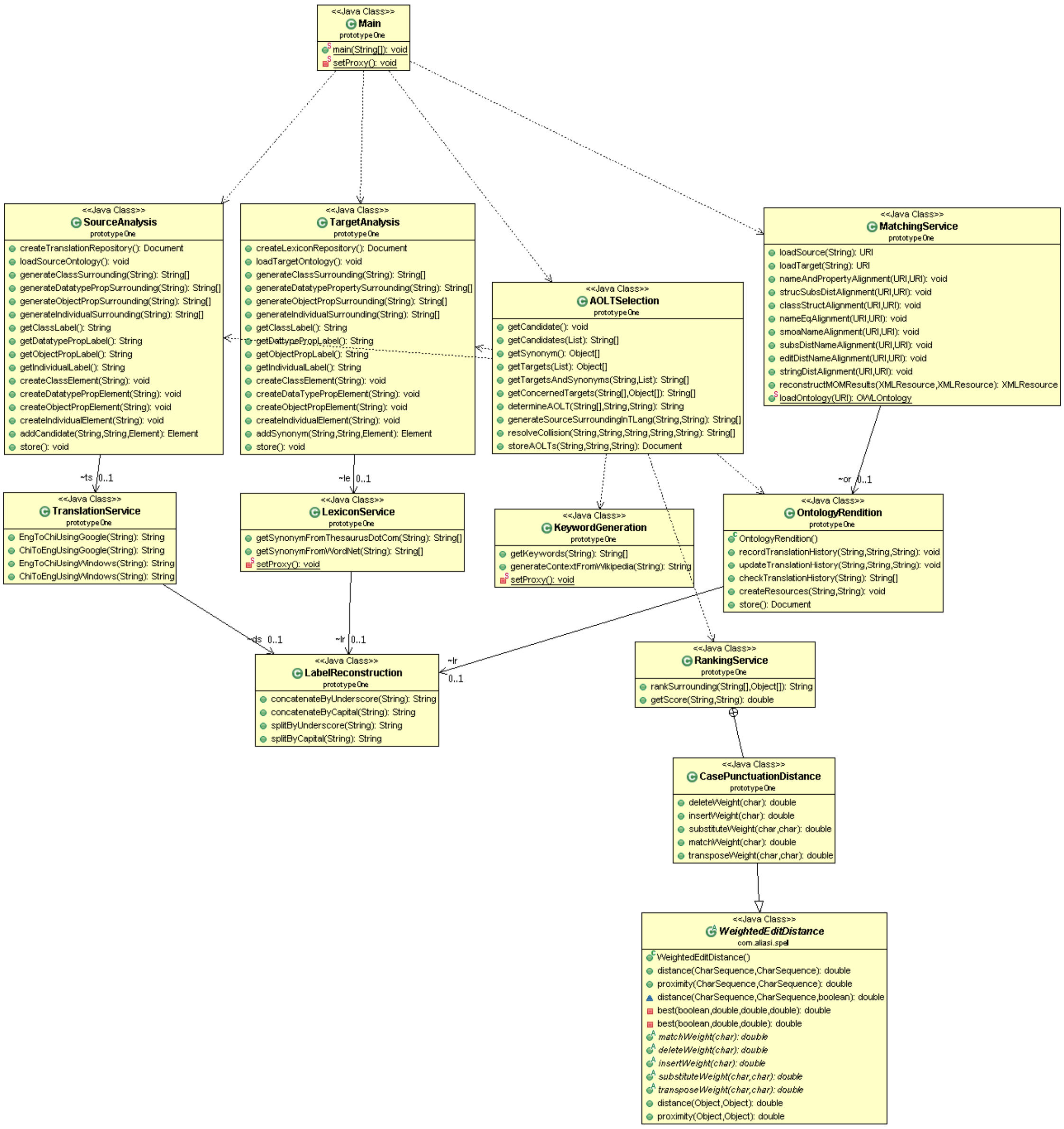


Figure 4-7. UML Class Diagram of SOCOM

4.5. SOCOM Evaluation

The basic AOLT process is evaluated through the evaluation of the cross-lingual ontology matching results generated by SOCOM. Two CLOM experiments were carried out in the evaluation. CLOM evaluations rely on multilingual ontologies and accompanying gold standards. Although multilingual ontologies are easy to come by, however, complete (as opposed to partial gold standards, discussed in chapter 2, section 2.4.1) and readily available gold standards are difficult to find. As a result, in this section, one experiment (discussed in section 4.5.2) uses third party developed multilingual ontologies and gold standard, whereas the other experiment (discussed in section 4.5.1) uses a manually generated ontology and gold standard. Section 4.5.1 presents the setup and the findings of the first experiment involving ontologies labelled in Chinese and English. These ontologies in experiment one contain overlapping domains regarding the research community, and differ greatly in structure. The natural languages in them are examples of natural language pairs from different natural language families (i.e. the Chinese language is of the Sino-Tibetan language family, the English language is a Germanic language which is a subdivision of the Indo-European language family). Section 4.5.2 presents the setup and the findings of a second experiment involving ontologies labelled in English and French. The ontologies in experiment two concern the bibliography domain, and are much more similar to each other in comparison to the ontologies in experiment one. They not only contain highly similar domain coverage and structures, the natural languages in them are examples of natural languages from the same language family (i.e. the English language is a Germanic language and the French language is a Romance, which are subdivisions of the same natural language family: Indo-European languages).

In the CLOM experiments presented in this chapter, SOCOM is evaluated and compared to the baseline system. The baseline system (discussed in chapter 3, section 3.3) uses the GoogleTranslate API to achieve ontology label translations during ontology rendition, and the Alignment API to generate MOM results (between the rendered ontology and the target ontology) which are finally reconstructed to CLOM results (based on the ontology label translations and the MOM results). SOCOM (discusses previously in section 4.4) draws from within (e.g. semantic surroundings of the ontological resources at hand) and background information (e.g. synonyms of target ontology labels) to achieve *appropriate* ontology label translations during ontology rendition, and uses the same API and technique to generate the final CLOM results. As the only difference between these two systems is the ontology label translation, the experiments thus evaluate the proposed basic AOLT process exclusively. Such an experimental setup eliminates other contributors (such as both systems were implemented by the thesis author) that may potentially bias the evaluation findings, since the only variable is how the translations were achieved. The metrics used in the evaluations presented in this chapter consist of the measures identified in chapter 2 (section 2.7). Precision, recall and f-measure scores are generated as indicators of the quality of the matches created by the baseline system and SOCOM. Means and standard deviations are used to evaluate the confidence levels of the matches generated. Additionally, statistical analysis in the form of two-tailed paired t-tests is carried out on the f-measure scores collected to validate the statistical significance of the findings.

4.5.1. Experiment One

This section presents the experimental setup and the findings from a CLOM scenario involving ontologies labelled in Chinese and English, which are examples of ontologies with natural languages from different language families and containing overlapping domains of interest. SOCOM is compared against the baseline system through the evaluation on the matches generated by both systems. The remainder of this section is organised as follows. Section 4.5.1.1 presents the setup of the CLOM experiment and section 4.5.1.2 presents the findings and analysis from this evaluation.

4.5.1.1. Experimental Setup

The goal of this experiment is to evaluate and compare the mapping quality of the two CLOM systems in a scenario involving ontologies with natural languages from different language families, overlapping domains and different structures. This experiment uses the CSWRC (in Chinese, created based on the SWRC ontology) and the ISWC ontology (in English) describing the domain of the research community.

The SWRC and the ISWC ontology were first introduced in chapter 3 (section 3.4) which are both labelled in English. Based on the SWRC ontology, a team of domain experts (excluding the author of this thesis) manually developed the CSWRC[71] ontology using the Protégé editor. Note that the CSWRC ontology differs from the HSWRC ontology (discussed in chapter 3, section 3.4.1.1) which was generated by the author of this thesis. The CSWRC ontology is used here (as opposed to the HSWRC ontology) because it is generated independent of this author and is free from author intervention. The experiment presented in this section requires the CSWRC ontology to be a reliable version of the SWRC ontology (as opposed to just one rendition of the

[71] http://www.scss.tcd.ie/~bofu/SOCOMExperimentJuly2009/Ontologies/CSWRC.owl

SWRC ontology as in the case of the HSWRC ontology), as the gold standard is generated (by a separate group of experts) based on mappings from the SWRC ontology to the ISWC ontology (discussed later in this section). The CSWRC ontology is essentially the SWRC ontology that has been re-labelled in Chinese while retaining the same structure. There are 54 classes, 44 object properties and 30 data type properties in the CSWRC ontology. The creators of the CSWRC ontology are two full-time computer science researchers, one holds a doctoral degree in computer science and the other is a Ph.D. candidate in computer science. Both have knowledge and experience of ontologies and are native speakers of Chinese. As they are both researchers, they were familiar with the concepts in the domain that the SWRC ontology covers, and so there can be confidence in their translations. The process of creating the CSWRC ontology and is illustrated in figure 4-8.

As shown in figure 4-8, each expert was given a copy of the English SWRC ontology, which was then loaded into the Protégé editor. Each expert independently worked through the ontological entities shown in the editor and renamed them in Chinese. In order to keep a record of the renamed terms for later discussion, entities were renamed with their original English labels attached with Chinese labels. For instance, a class originally labelled as *Department* in the SWRC ontology is renamed as *Department_部门*. After each expert had independently completed this renaming process, further discussions were carried out by the team concerning the entities that had been given differing labels until both experts came to a consensus on the most suitable choice for all the renamed labels in the ontology. This discussion was facilitated by the author but the author did not participate. Finally, the CSWRC ontology was generated using new namespaces and the set of agreed Chinese labels for the named entities in the ontology. The CSWRC ontology contains the same semantics (i.e. structured conceptualisations in the same way) as the SWRC ontology, except that all of its entities are labelled in Chinese. The CSWRC ontology can be found at `root/SOCOMExperiments/ExperimentOne/Ontologies` on the DVD.

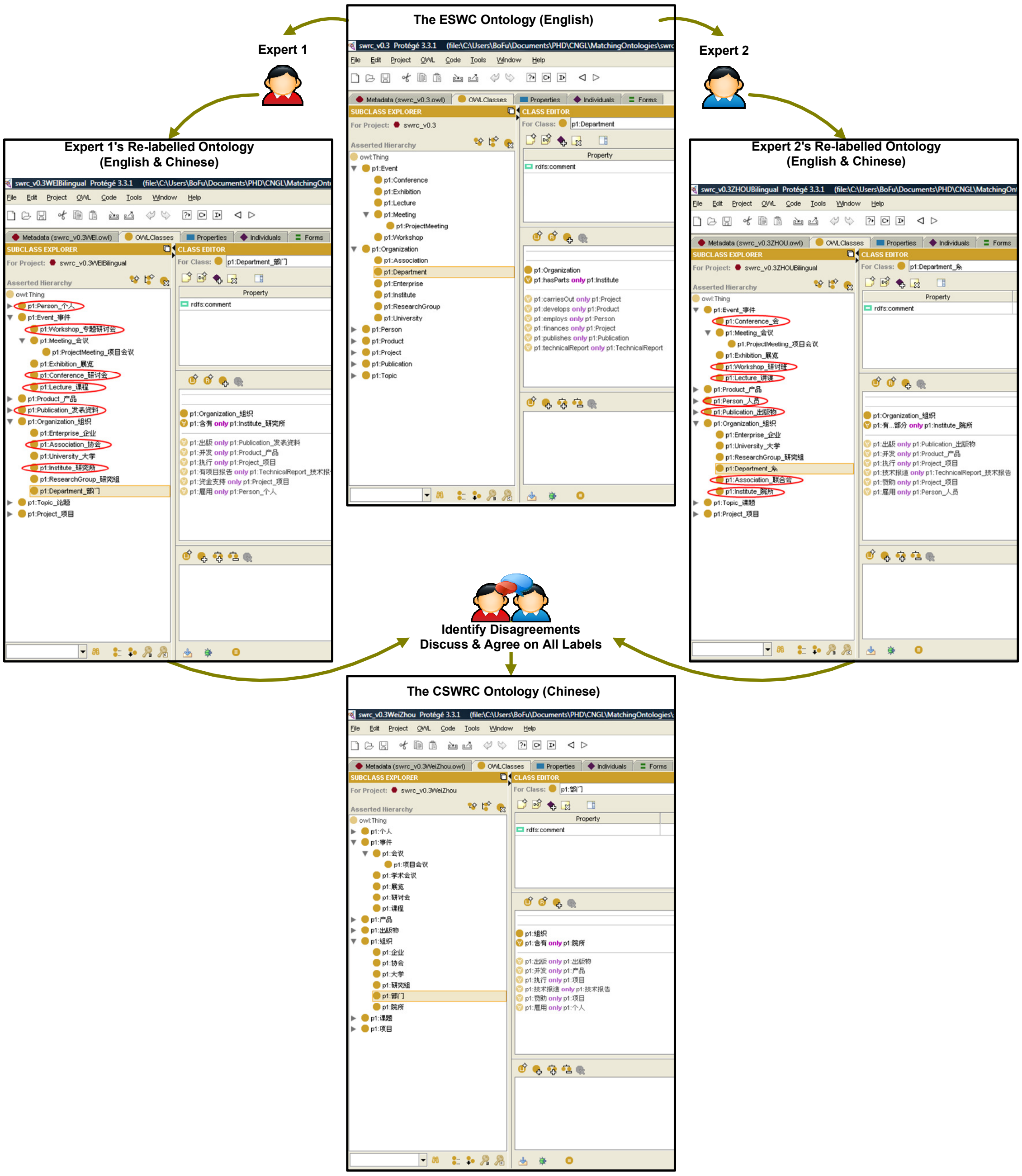


Figure 4-8. The Creation Process of the CSWRC Ontology

The evaluation of the two CLOM systems relies on the availability of a set of reliable mappings between the Chinese and English ontologies, i.e. a gold standard. The CSWRC ontology is viewed as a semantic equivelant of the SWRC ontology in the experiments since the conceptualisations are structured in the same way (although different natural languages are used for the labelling of the concepts), so that the gold standard between the SWRC ontology and the ISWC ontology is in fact also the gold standard between the CSWRC ontology and the ISWC ontology. To minimuse bias, the group of experts who created the CSWRC ontology is different with the groups of experts who established the gold standard between the SWRC ontology and the ISWC ontology. As the translations were recorded for the original SWRC concepts, the gold standard between the SWRC ontology and the ISWC otnology can thus be referred to the corresponding concepts in the CSWRC ontology.

The gold standard between the SWRC ontology and the ISWC ontology was created as follows. A total of seven ontology mapping experts (excluding the creators of the CSWRC ontology) were selected to validate the mapping standard in order to minimise any partial judgment. Among these experts, two hold Ph.D.s in computer science, and the others are Ph.D. candidates including the author of this thesis. A first version of the mapping standard was created (by the author of this thesis) between the SWRC ontology and the ISWC ontology, which was then passed onto each of the six remaining members. Each expert then independently examined the tentative mappings and highlighted the doubtful mappings for further discussion. Finally, a meeting was held to discuss the matches that were questionable until an agreed set of mappings for the gold standard was concluded. As the alignment API only generated one-to-one exact matches in the experiments, the experts concentrated on the validation of exact matches for the gold standard (i.e. resources that contain semantically equivalent labels with the same domain and range specifications should they be declared) between the SWRC ontology and the ISWC ontology in the meeting. The final gold standard (between Chinese entities in the CSWRC ontology and English entities in the ISWC ontology) includes 41 exact matches between the CSWRC ontology and the ISWC ontology, which can be found at `root/SOCOMExperiments/ExperimentOne/GoldStandard/` on the DVD. The mapping procedures carried out in the experiment can be illustrated by figure 4-9.

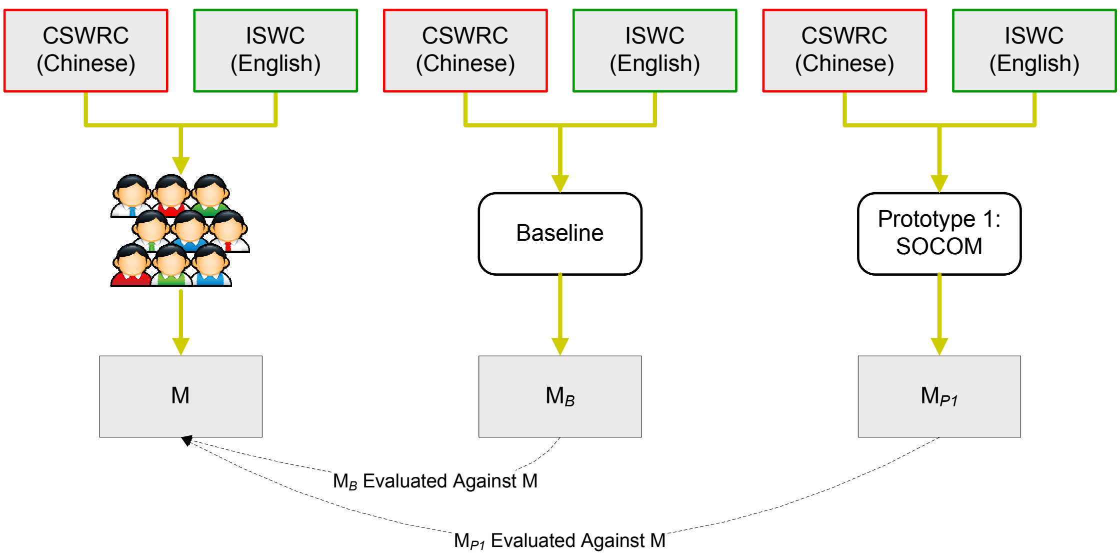


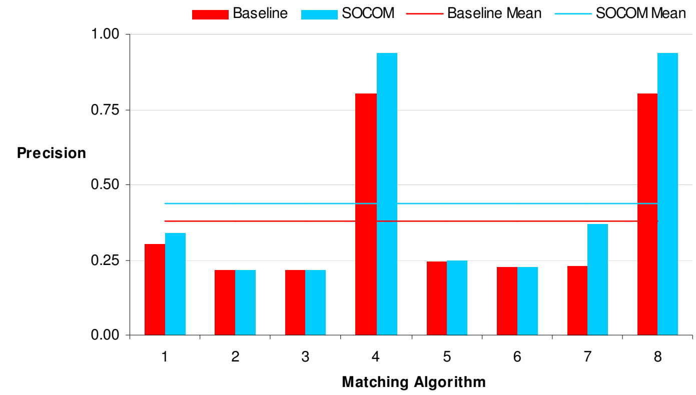
Figure 4-9. Experiment One Overview

As shown in figure 4-9, the gold standard when mapping the CSWRC ontology to the ISWC ontology is M, which was generated by a group of experts. Two CLOM systems, namely the baseline system and SOCOM are each executed to generate mappings between these ontologies as M_B and M_{P1} respectively. The quality (in terms of precision, recall and f-measure) of M_B and M_{P1} are finally calculated with respect to M to determine their precision, recall and f-measure. Eight MOM algorithms (supported by the Alignment API) are applied in both CLOM systems, which means eight sets of matches (as each MOM algorithm creates its own set of matches) are included in M_B (see `root/SOCOMExperiments/ExperimentOne/Mappings/MB/` on the DVD), and another eight sets of matches are included in M_{P1} (see `root/SOCOMExperiments/ExperimentOne/Mappings/MP1/` on the DVD). Recall findings shown in chapter 3 (section 3.4): different mapping outcome (i.e. varied precision, recall and f-measure) were generated depending on the actual MOM algorithm applied in the baseline system, it is thus of interest to apply the same algorithms and investigate whether they will improve given the SOCOM system with the AOLT process.

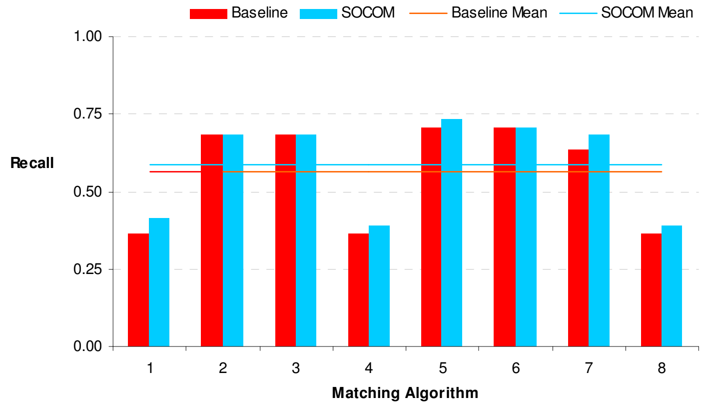
4.5.1.2. Findings and Analysis

This section presents the findings and analysis of experiment one. In particular, the precision, recall and f-measure of M_B and M_{P1} are calculated and compared against each other. In addition, confidence levels of the matches in them are examined and compared. Lastly, paired t-test is used to validate the statistical significance of the results. Raw data from this experiment can be found at `root/SOCOMExperiments/ExperimentOne/Evaluation/` on the DVD.

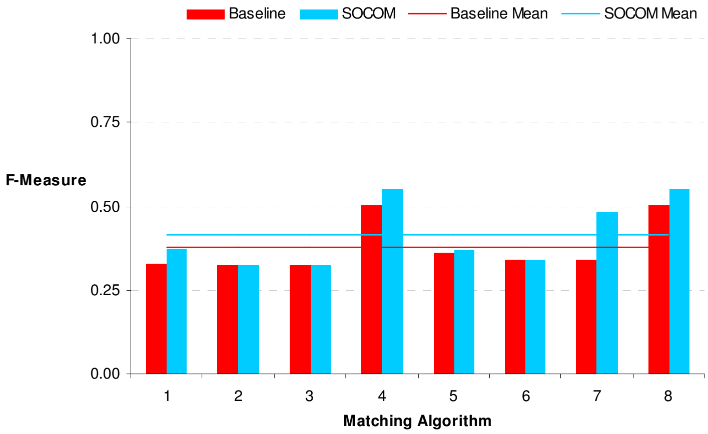
The precision (figure 4-10-a), recall (figure 4-10-b) and f-measure (figure 4-10-c) found for M_B and M_{P1} are shown in figure 4-10. These results are calculated on the basis that a match is considered correct as long as it is included in the gold standard M, regardless of its confidence level. The x-axis in figure 4-10-a presents the sets of matches generated by the eight MOM algorithms, and the y-axis presents the precision found for these match sets. For example, when applying the *EditDistNameAlignment* algorithm, M_B generated less than 0.25 precision while M_{P1} generated just below 0.38 precision. Similarly, the x-axis in figure 4-10-b illustrates sets of matches generated and the y-axis illustrates the recall found in them. Finally, f-measure scores are shown in figure 4-10-c with the x-axis illustrating sets of matches generated and the y-axis illustrating the f-measure. The charts in figure 4-10 also include precision, recall and f-measure means for both M_B and M_{P1}, which visually illustrate the averages (across eight algorithms in terms of precision, recall and f-measure) found in both systems and present an overview for the improvements gained by SOCOM.



(a) Precision



(b) Recall



(c) F-Measure

Legend:			
1	NameAndPropertyAlignment	5	SMOANameAlignment
2	StrucSubsDistAlignment	6	SubsDistNameAlignment
3	ClassStructAlignment	7	EditDistNameAlignment
4	NameEqAlignment	8	StringDistAlignment

Figure 4-10. Experiment One Precision, Recall and F-Measure Results

Figure 4-10-a shows that all eight matching algorithms indicate equal (in the case of the *StrucSubsDistAlignment* algorithm, the *ClassStructAlignment*[72] algorithm and the *SubsDistNameAlignment* algorithm) or higher (in the case of the *NameAndProperty-Alignment* algorithm, the *NameEqAlignment* algorithm, *SMOANameAlignment* algorithm, the *EditDistNameAlignment* algorithm and the *StringDistAlignment* algorithm) precision when using SOCOM than when using the baseline system. This finding demonstrates that SOCOM was able to generate at least the same number of correct matches if not more than the baseline system. A precision mean of 0.4367 is found in M_{P1}, whereas a mean of 0.3793 is found in M_B. This is an average improvement of 15.13% on precision, which indicates that overall, SOCOM generated more correct matches than the baseline system in this experiment.

Figure 4-10-b shows a similar trend for the recall achieved in M_B and M_{P1}. The matching techniques used generated equal (in the case of the *StrucSubsDistAlignment* algorithm, the *ClassStructAlignment* algorithm and the *SubsDistNameAlignment* algorithm) or higher (in the case of the *NameAndPropertyAlignment* algorithm, the *NameEqAlignment* algorithm, *SMOANameAlignment* algorithm, the *EditDistName-Alignment* algorithm and the *StringDistAlignment* algorithm) recall when using SOCOM than when using the baseline system. This finding suggests that the matches generated by SOCOM were at least as complete as the ones generated by the baseline system. The mean of recall scores at 0.5854 was found in M_{P1}, whereas a lower recall mean of 0.5640 was found in M_B. This is an average improvement of 3.79% on the completeness of the matches generated when using SOCOM.

Taking both precision (i.e. the correctness of the matches generated) and recall (i.e. the completeness of the matches generated) into account, figure 4-10-c demonstrates the overall quality of the matches found in M_B and M_{P1} through the derived f-measure scores. Equal (in the case of the *StrucSubsDistAlignment* algorithm, the *ClassStruct-Alignment* algorithm and the *SubsDistNameAlignment* algorithm) or higher (in the case of the *NameAndPropertyAlignment* algorithm, the *NameEqAlignment* algorithm, *SMOANameAlignment* algorithm, the *EditDistNameAlignment* algorithm and the *StringDistAlignment* algorithm) f-measure scores are found when using SOCOM in comparison to the baseline system, which suggests SOCOM is able to generate matches of at least equal quality if not higher than

[72] In all experiments shown in this chapter, the ClassStructAlignment algorithm is accompanied by the SMOANameAlignment algorithm to generate matches as it only works with another algorithm.

the baseline system. A mean f-measure score of 0.4146 was found in M_{P1}, and a lower mean f-measure was found in M_B as 0.3782. This is an average improvement of 9.62% on the f-measure, suggesting that the matches are of a higher quality when using SOCOM than when using the baseline system.

It may be argued that since the differences of the f-measure shown in figure 4-10-c between the baseline system and SOCOM are moderately small, it may be insufficient to conclude a difference between the two systems. To validate the statistical significance of the findings, paired t-tests were carried out on the f-measure scores collected across eight matching algorithms, where a p-value of 0.007 was found. The null hypothesis of this paired t-test is that there is no difference between the baseline system and SOCOM, at a significance level of $\alpha=0.05$, the p-value rejects the null hypothesis. This result supports what has been indicated by the findings: the matches generated by SOCOM is of higher quality.

To evaluate the confidence levels of the matches generated, the means and the standard deviations of the confidence levels accompanying the matches in M_B and M_{P1} are examined. Figure 4-11 presents a scattered plot of the data collected in table 4-2, which aims to provide visual assist with the understanding of these data. The standard deviations are presented by the x-axis and the confidence means are presented by the y-axis. Higher quality matches are those dotted in the area of the top left corner on the graph (i.e. high confidence mean and low deviation) as opposed to those dotted at the bottom right corner (i.e. low confidence mean and high deviation).

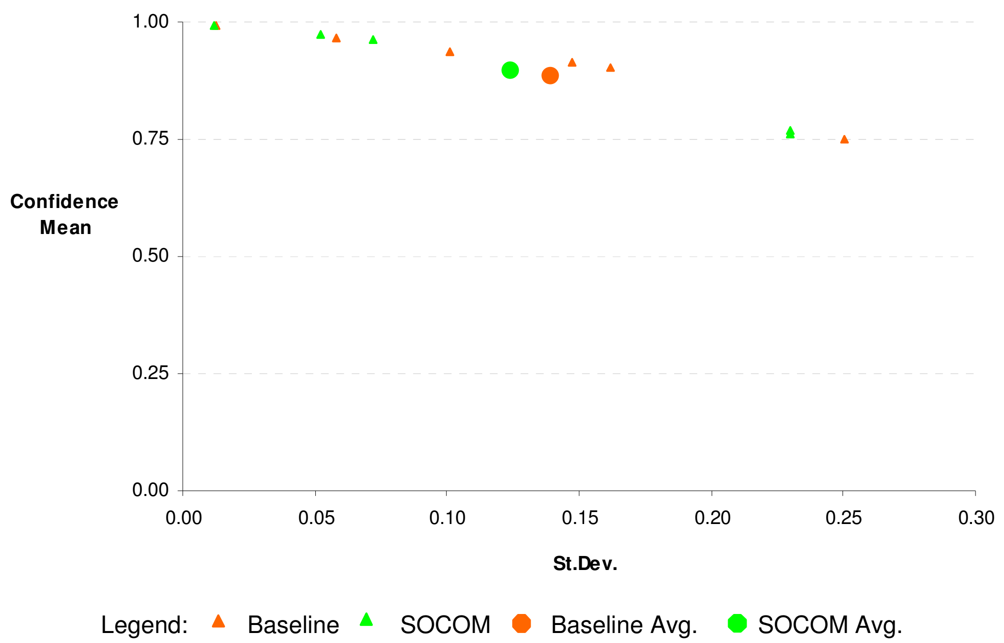


Figure 4-11. Evaluation on Confidence Levels - Experiment One

Table 4-2. Experiment One Confidence Data

Matching Algorithm	Baseline		SOCOM	
	St. Dev.	Mean	St. Dev.	Mean
1 NameAndPropertyAlignment	0.1014	0.9374	0.0718	0.9638
2 StrucSubsDistAlignment	0.2505	0.7505	0.2298	0.7682
3 ClassStructAlignment	0.2505	0.7505	0.2298	0.7630
5 SMOANameAlignment	0.0582	0.9649	0.0525	0.9723
6 SubsDistNameAlignment	0.1618	0.9041	0.1473	0.9133
7 EditDistNameAlignment	0.0123	0.9909	0.0119	0.9914
Avg.	**0.1391**	**0.8830**	**0.1239**	**0.8962**

In figure 4-10, the orange triangles are data collected from the baseline system and the green triangles are data collected from SOCOM. Note that not all matching algorithms generate matches with varied confidence levels, for instance, the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm only created matches that have a confidence level of 1.0 in this experiment, hence they are not included in figure 4-10. As shown in table 4-2, for every matching algorithm that did generate matches with varied confidence, the mean has been increased when using SOCOM. In addition, the standard deviations of all confidence levels have been found to be decreased when using SOCOM. The mean (i.e. a point with average standard deviation on x-axis and average confidence mean on y-axis for each system) for the baseline system is presented by an orange dot, and the average dot for SOCOM is represented by a green dot in figure 4-10. It is visibly shown that the green dot is of higher mean and lower standard deviation. On average, there is an improvement of 1.49% on the confidence mean and a reduction by 10.93% on the standard deviation when comparing SOCOM to the baseline system. This finding consistently indicates that matches generated using SOCOM are of higher quality than those generated using the baseline system, because they are not only more confident but also less dispersed.

The findings from the first experiment have been positive. However, as the ontologies used in this experiment contain overlapping domains and unrelated natural language families. It may be argued that this is a scenario where the AOLT process is most likely to be beneficial, which raises the question of how well will SOCOM work with ontologies containing highly similar semantics and closely related natural languages. This is investigated in a second experiment, discussed next.

4.5.2. Experiment Two

This section presents the experimental setup and the findings from a second CLOM scenario involving SOCOM. The ontologies used in the second experiment are labelled in English and French, of the bibliography domain and with highly similar structures. In contrast to the first experiment, this second experiment concerns ontologies that not only contain highly similar semantics (i.e. structured conceptualisations) but also involve natural languages of the same family. The remainder of this section is organised as follows. Section 4.5.2.1 presents the setup of the experiment and section 4.5.2.2 presents the findings and analysis from this evaluation.

4.5.2.1. Experimental Setup

Figure 4-11 gives an overview of the experimental process. The ontologies (shown as ontology 101[73] and ontology 206[74]) and the evaluation gold standard[75] (shown as M') used in this experiment are taken from the Benchmark datasets from the Ontology Alignment Evaluation Initiative (OAEI) 2009 campaign. The 101 ontology is labelled in English, consists of 36 classes, 24 object properties, 46 data type properties and 137 instances. Ontology 206 contains almost the same semantics (i.e. conceptualisations and how they are structured), except it has one less object property and is labelled in French. More specifically, ontology 206 is in French, consists of 36 classes, 23 object properties, 46 data type properties and 137 instances. The gold standard (between English entities in the 101 ontology and French entities in the 206 ontology) provided by the OAEI contains 97 exact matches between the 101 and the 206 ontology. The benchmark dataset and the gold standard were one of the first introduced by OAEI since its establishment in 2004. Over the years, variations of the datasets were introduced and the gold standards had been updated accordingly. These gold standards were generated by the OAEI organisers who are experts on ontology mapping.

In the second experiment, the baseline system and SOCOM are each applied to generate mappings (shown as M_B' and M_{P1}' in figure 4-12; M_B' can be found at `root/SOCOMExperiments/ExperimentTwo/Mappings/MBPrime/` and M_{P1}' can be found at `root/SOCOMExperiment/ExperimentTwo/Mappings/MP1Prime/` on the DVD) between the 101 and the 206 ontology, and the quality of the mappings are evaluated against the gold standard M'. The goal of this experiment is to further investigate the effectiveness of the AOLT process when working with ontologies that are very similar to each other.

[73] http://oaei.ontologymatching.org/2009/benchmarks/101/onto.rdf
[74] http://oaei.ontologymatching.org/2009/benchmarks/206/onto.rdf
[75] http://oaei.ontologymatching.org/2009/benchmarks/206/refalign.rdf

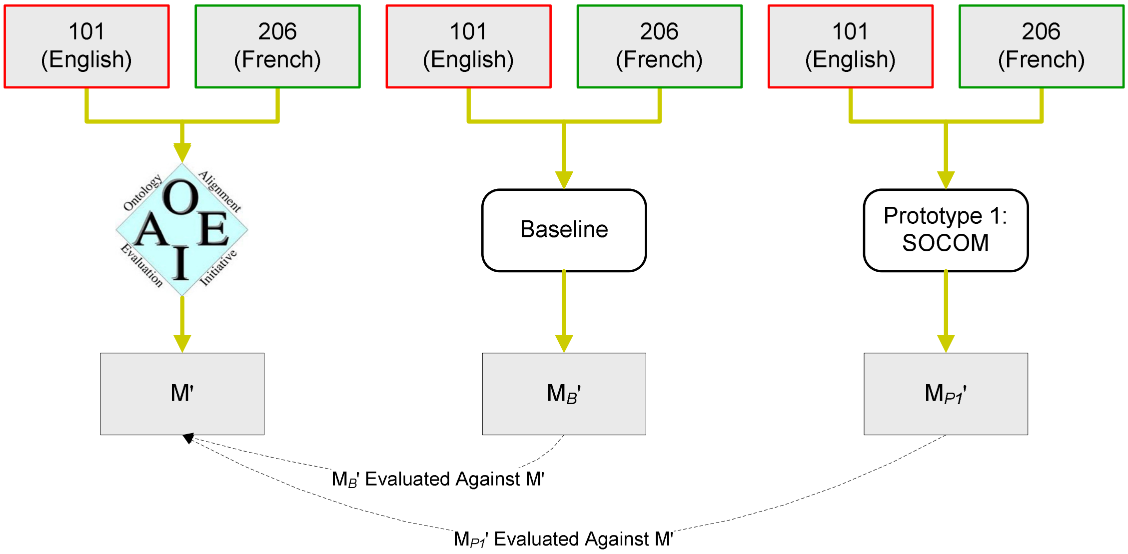
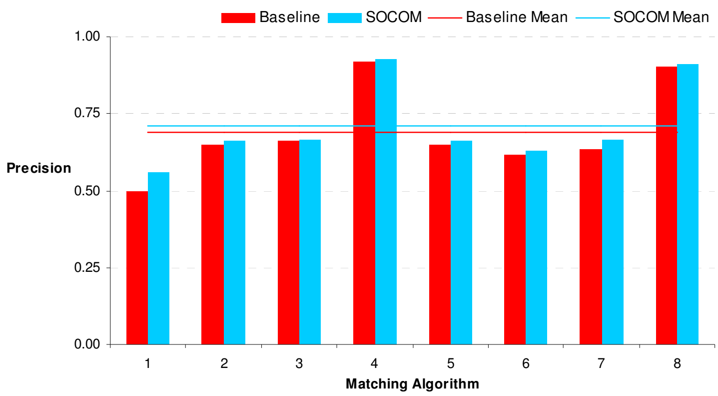


Figure 4-12. Experiment Two Overview

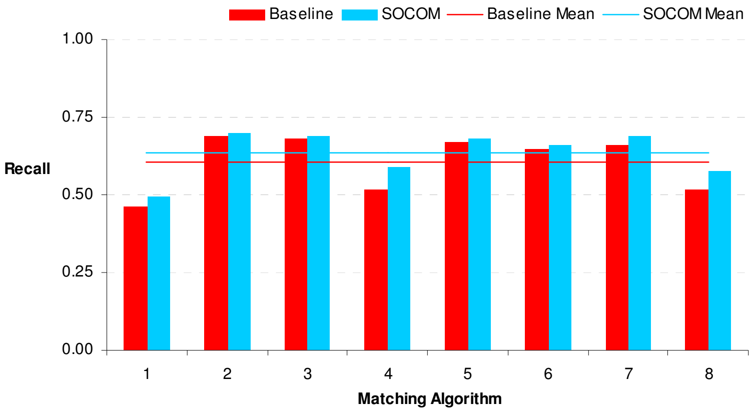
The original OAEI test case aims to assess the effectiveness of structure-based MOM techniques, since the 101 and the 206 ontology are highly similar in structure and contain almost identical domain knowledge. Note that translations of ontology labels were not included in the original OAEI test case. Nevertheless, this scenario satisfies the requirement for a second CLOM experiment, considering ontologies with similar semantics and natural languages from are presented with a reliable gold standard.

4.5.2.2. Findings and Analysis

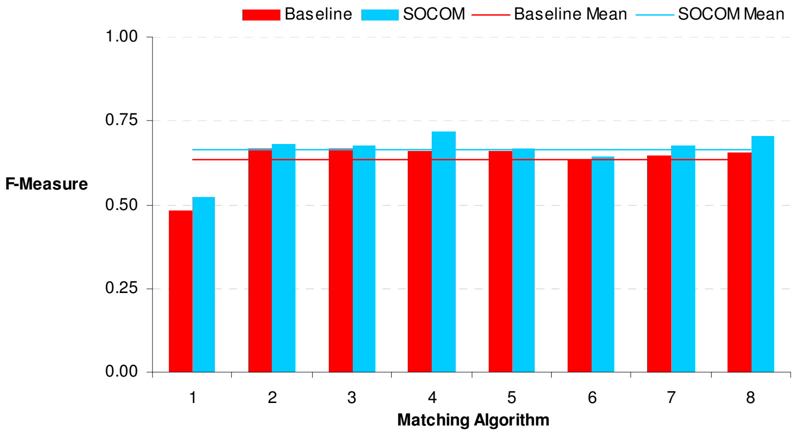
The precision (figure 4-13-a), recall (figure 4-13-b) and f-measure (figure 4-13-c) scores found in M_B' and M_{P1}' are shown in figure 4-13. These results are calculated based on comparisons made to M', and a correct match is one that is included in M' regardless of its confidence level. Raw data can be found at `root/SOCOMExperiments/ExperimentTwo/Evaluation/` on the DVD.



(a) Precision



(b) Recall



(c) F-Measure

Legend:				
1	NameAndPropertyAlignment	5	SMOANameAlignment
2	StrucSubsDistAlignment	6	SubsDistNameAlignment
3	ClassStructAlignment	7	EditDistNameAlignment
4	NameEqAlignment	8	StringDistAlignment

Figure 4-13. Experiment Two Precision, Recall and F-Measure Results

Figure 4-13-a shows improved precision scores for all matching algorithms when SOCOM is applied, indicating that a greater number of correct matches were found when using SOCOM than using the baseline system. An average precision of 0.6918 was found in M_B', whereas 0.7084 was found in M_{PI}'. This is an average improvement of 2.40% on precision. Greater improvements can also be seen on the recall of the matches generated when using SOCOM. As figure 4-13-b shows, more completed matches were found in every matching technique that was accompanied by SOCOM. An average recall of 0.6057 was found in M_B', whereas 0.6353 was found in M_{PI}'. This is an average improvement of 4.89% on the recall of the matches generated. With improved precision and recall, the f-measure of M_{PI}' for each matching algorithm are increased as shown in figure 4-13-c. An average f-measure of 0.6347 was found in M_B', whereas 0.6621 was found in M_{PI}'. This is an average improvement of 4.32% on the overall quality of the matches generated when applying SOCOM.

Compared to the improvements shown in the first experiment, the improvements on precision, recall and f-measure scores appear smaller in the second experiment. To validate the statistical significance of the difference (if there is a difference) between the two systems in the second experiment, paired t-test was carried out on the f-measure scores, and a p-value of 0.008 was found. At a significance level of $\alpha=0.05$, this finding supports the hypothesis of there being a difference between the baseline system and SOCOM in this experiment. This further indicates with confidence that SOCOM generated higher quality matches than the baseline approach in this second experiment.

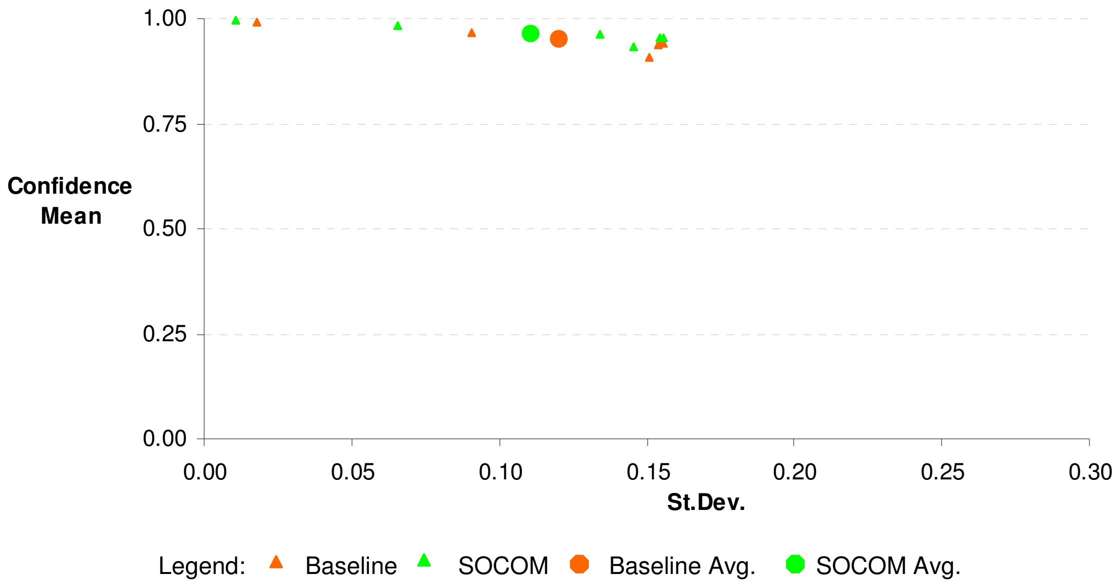


Figure 4-14. Evaluation on Confidence Levels - Experiment Two

Table 4-3. Experiment Two Confidence Data

Matching Technique	Baseline		SOCOM P.1	
	St. Dev.	Mean	St. Dev.	Mean
1 NameAndPropertyAlignment	0.0909	0.9674	0.0659	0.9832
2 StrucSubsDistAlignment	0.1509	0.9059	0.1454	0.9311
3 ClassStructAlignment	0.1545	0.9440	0.1543	0.9557
5 SMOANameAlignment	0.1556	0.9431	0.1554	0.9551
6 SubsDistNameAlignment	0.1541	0.9372	0.1344	0.9621
7 EditDistNameAlignment	0.0179	0.9913	0.0108	0.9968
Avg.	**0.1207**	**0.9481**	**0.1110**	**0.9640**

Figure 4-14 shows the confidence means and standard deviations of the matches in M_B' (in orange) and M_{P1}' (in green). The data used to generate the plot in figure 4-14 is shown in table 4-3. The subjects for this study are those correct matches with varied confidence levels, which excludes those algorithms that only generate matches with 1.0 confidence levels (this is the same with experiment one). On average, the standard deviation of the confidence levels in M_B' is 0.1207 and the confidence mean is 0.9481. This result is improved in M_{P1}', with an 8.04% decrease on standard deviation (at 0.1110) and a 1.68% increase on confidence mean (at 0.9640). This improvement is shown visibly in figure 4-14 from the positioning of the orange (the M_B' mean) and the green (the M_{P1}' mean) dots. This finding supports previous evidence and further demonstrates that the matching quality is higher when the AOLT process is deployed.

4.5.3. Conclusions

SOCOM was evaluated against the baseline system in two CLOM experiments. The first experiment concerned ontologies with natural languages from different language families, different structures and overlapping domains. The second experiment involved ontologies of the same domain with natural languages from the same language family, as well as highly similar concepts and structures. The evaluation results from both experiments show improvements in matching quality when SOCOM was applied. A summary of the average improvements found in the two experiments is presented in table 4-4. The improvement shown in this thesis is calculated as follows. Given systems *A* and *B*, and their results R_A and R_B respectively, the improvement of *B* with respect to *A* is $(R_B-R_A)/R_A$. In table 4-4, these improvements are shown as percentage, which can be + (i.e. an increase) or – (i.e. a decrease). It can be arugued that when synonyms are generated, additional ambiguity may be introduced to the translation selection process. However, it is not of interest to measure ambiguity during the translation process as this research is an investigation of the quality of the matching results that were generated using translation-based cross-lingual matching processes.

Table 4-4. Key Findings in Experiment One and Two

Findings		Baseline	SOCOM	Improvement (%)
Exp. 1	Precision	0.3793	0.4367	15.13
	Recall	0.5640	0.5854	3.79
	F-Measure	0.3782	0.4146	9.62
	Confidence Level Mean	0.8830	0.8962	1.49
	Confidence Level St.Dev.	0.1391	0.1239	-10.93
Exp. 2	Precision	0.6918	0.7084	2.40
	Recall	0.6057	0.6353	4.89
	F-Measure	0.6347	0.6621	4.32
	Confidence Level Mean	0.9481	0.9640	1.68
	Confidence Level St.Dev.	0.1207	0.1110	-8.04

Proportionally speaking, precision has been improved by a greater extent in experiment one than in experiment two, whereas the opposite is found on the improvement regarding recall. This finding suggests that, when dealing with ontologies with diverse characteristics (i.e. natural languages from different language families, varied structures and overlapping domains), the improvement on the precision may be more visible than the improvement on the recall when applying the AOLT process. On the other hand, when dealing with ontologies with very similar characteristics (i.e. natural languages from the same language family, similar structures and domains), the improvement on the recall may be more evident than the precision when applying the AOLT process. In both experiments, the improvements on the matching confidence (see standard deviation and mean scores in table 4-4) have been relatively similar, suggesting more confident and less dispersed matches can be generated using the AOLT process despite vast variance in ontology characteristics.

The evaluation presented in this section is somewhat limited in its number of ontologies and the natural language pairs experimented with. However, these experiments are representative of CLOM scenarios that involve distinct and similar ontology characteristics as well as natural languages, thus offer this research with a preliminary finding: there is a noticeable potential of the AOLT-based cross-lingual ontology mapping. In particular, the experimental findings successfully validate the soundness of the AOLT concept. Since a basic implementation of the AOLT process has been proven to be effective, it is motivating to investigate whether a more sophisticated AOLT process would be more effective at improving CLOM quality. For instance, the AOLT result pool can be increased (e.g. by generating synonyms of the candidate translations of the O_1 labels) to allow more candidate AOLT results to be selected for a given O_1 label. This could also increase the number of available alternative AOLT results should the initial AOLT result cause collision. It is shown through the evaluations of SOCOM that depending on the translations selected for the source labels, the mapping quality consequently differ (i.e. higher precision, recall and f-measure were found in SOCOM than in the baseline system). Given various candidate AOLT results, one way to influence the mapping outcome is to alter the AOLT results for the labels in the given source ontology. Incorporating additional inputs to influence the AOLT outcome motivates the key research direction for the second prototype: SOCOM++ (discussed in chapter 5).

4.6. Case Study

Motivated by the positive preliminary findings from the evaluation of SOCOM, and particularly its support for generating mappings that are carried out in the multilingual environment, the author of this thesis was encouraged to apply the SOCOM system to a wider application that could potentially benefit from the use of CLOM techniques.

Such an opportunity arose within the context of Science Foundation Ireland[76] funded the Centre for Next Generation Localisation[77] (CNGL) project. SOCOM is applied to the Adaptive Retrieval and Composition of Heterogeneous Information sources for personalised hypertext Generation (ARCHING) system [Steichen et al., 2011] in this case study. The ARCHING system is a cross-lingual information retrieval (CLIR) system specialising in the customer support domain for Norton 360[78] - a home security product from one of CNGL industrial partners: Symantec[79].

A vast amount of customer support documents – structured (e.g. enterprise technical documentations in XML, RDF, etc.) and unstructured (e.g. user generated content such as online forum posts in plain text form) – are often available in English but not in other natural languages such as German. It is thus of interest to seek ways to support information sharing across natural language barriers, so that the information in English can be accessed by German speakers (who also understands English) through the ARCHING system. The novelty of this case study is the application of cross-lingual ontology mapping techniques such as the SOCOM system in a CLIR system. The remainder of this section is organised as follows. Section 4.6.1 discusses the objectives and scopes of the case study. Section 4.6.2 presents some related background. The technical approaches undertaken in this study are discussed in section 4.6.3. Finally, section 4.6.4 discusses the significance of the study.

4.6.1. Objectives and Scope of the Case Study

This section presents the objectives and scope of the case study. The objective of the case study is to apply cross-lingual ontology mapping techniques for the purpose of cross-lingual information retrieval. In particular, there are two specific objectives:

- demonstrate the feasibility of SOCOM in a real world application: ARCHING;
- investigate the potential benefits and drawbacks from using CLOM techniques for the purpose of CLIR.

---

[76] Science Foundation Ireland is the statutory body in Ireland responsible for funding academic researchers and research teams for the purpose of strategic scientific research. More information can be found at http://www.sfi.ie/

[77] The CNGL is an SFI funded academia-industry partnership with over one hundred researchers developing novel technologies addressing the key challenges in localisation and personalisation. More information can be found at http://www.cngl.ie/index.html

[78] http://www.norton.com

[79] http://www.symantec.com/index.jsp

This case study concerns ontologies of the Norton 360 domain (i.e. conceptualisations related to product features and how-tos that are specific to the Norton 360 product). The natural languages of these ontologies include English and German. This study is not designed to be an exhaustive list of CLIR scenarios, but rather an example of CLIR that is achieved through the use of cross-lingual ontology mapping. This is a proof-of-concept study, aiming to validate the possibility of applying cross-lingual ontology mapping techniques in the context of CLIR.

4.6.2. Background of the Case Study

This section presents some related background regarding the application of ontology mapping techniques in information retrieval (IR) systems. The concept of using conceptual frameworks such as thesauri and ontologies in search systems [De Luca & Eul, 2007; Castells et al., 2007] for improved information access [Shuang et al., 2004] and enhanced user experiences [Stamou & Ntoulas, 2009] is well researched in the IR community. However, the use of ontology mapping as a technique to aid the search functions in IR has been relatively limited. The most advanced work of using ontology alignment in cross-lingual information retrieval (CLIR), to the best of this author's knowledge, is Zhang et al.'s statistical approach [Zhang et al., 2004] which does not involve translations of ontology labels. Instead, statistical approaches including latent semantic indexing[80], singular value decomposition[81], directed acyclic graphs[82] and maximal common subgraph[83] on document collections are applied. In order to apply Zhang et al.'s approach, parallel corpora must be generated beforehand so that statistical analysis can be carried out. However, this requirement can be a costly process: generating parallel corpora may not be possible or is computationally infeasible. In addition, statistical techniques (are applied to parallel corpora and) do not make use of

---

[80] Latent semantic analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text. [Landauer & Dumais, 1997] For an introduction on LSA, see [Landauer et al., 1998].

[81] Suppose M is an $m \times n$ matrix whose entries come from the field K, which is either the field of real numbers or the field of complex numbers. Then there exists a factorization of the form $M=U \bullet V^*$, where U is an $m \times m$ unitary matrix over K, the matrix $\bullet$ is an $m \times n$ diagonal matrix with nonnegative real numbers on the diagonal, and V^*, an $n \times n$ unitary matrix over K, denotes the conjugate transpose of V. Such a factorization is called the singular value decomposition (SVD) of M. For more information on SVD, see [Trefethen & Bau, 1997].

[82] A directed acyclic graph (DAG) is a directed graph with no directed cycles. See [Thulasiraman & Swamy, 1992] for more information.

[83] Given graphs and the objects within, to determine the degree and composition of the similarity between these objects, graph matching techniques are often applied. Graph matching can be formulated as a problem involving the maximum common subgraph (MCS) between the collection of graphs being considered [Raymond & Willett, 2002]. For more on MCS, see [Willett, 1999].

the existing semantic knowledge (e.g. the entities in the ontologies, how they are structured and related to one another) in the multilingual ontologies that are presented in an IR scenario. It is the view of this author that a novel approach to enable CLIR involving multilingual ontologies is to use cross-lingual ontology mapping techniques (e.g. the SOCOM system), whereby parallel corpora are not required and the embedded ontological semantics are accounted for. The core idea behind CLOM-enabled CLIR systems is illustrated by figure 4-15.

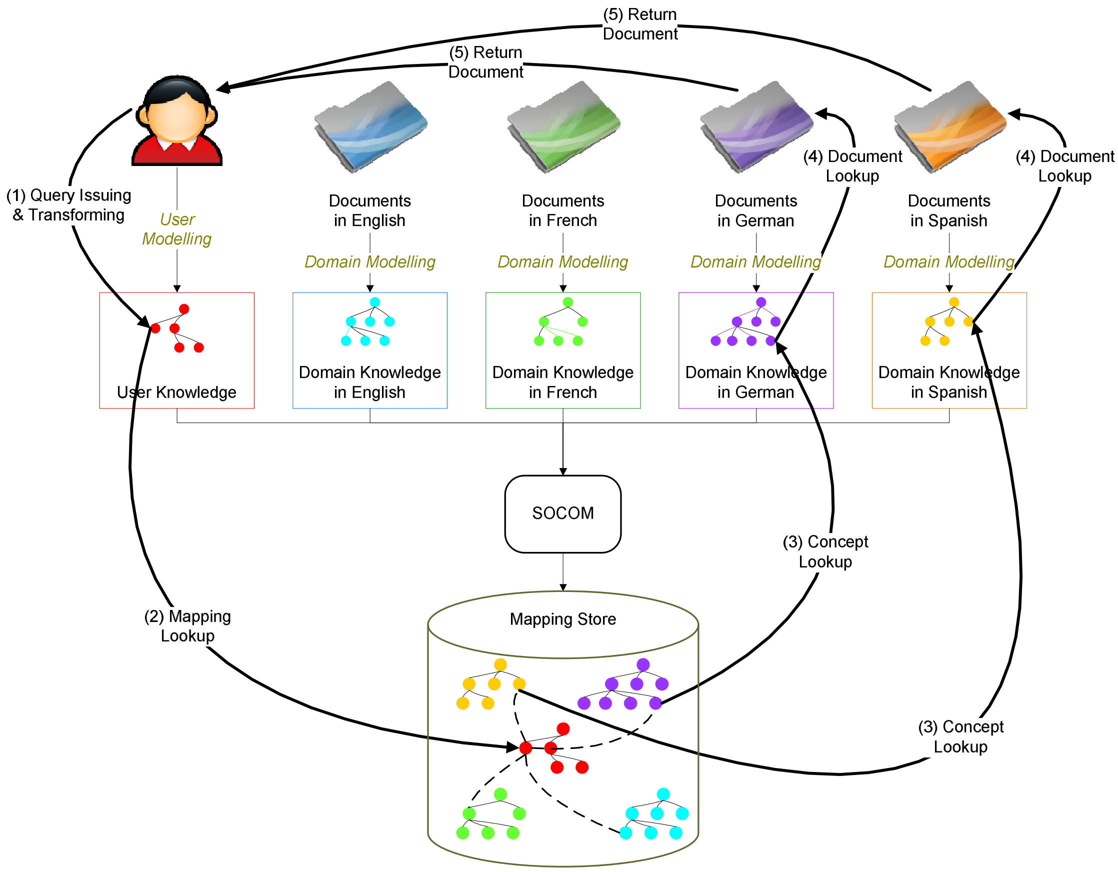


Figure 4-15. CLOM-Enabled CLIR

As shown in figure 4-15, given a bilingual or multilingual user of an IR system and a set of relevant documents in various natural languages, ontologies can be constructed through *user modelling* and *domain modelling* as structured models of the user and the resources pre-runtime. The user model (shown as *user knowledge* in figure 4-15) may contain information such as the user's interests and natural language preferences. The domain models (shown as *domain knowledge* in figure 4-15) in various natural languages may contain structured concepts of interests with associated documents as instances. To bridge between the user model and the domain models that are in a different natural language, SOCOM can be applied to generate mappings among these models (stored in the *mapping store* as shown in figure 4-15). At runtime, a query is issued by the user which can be associated to concept(s) in the user model (for example, by extracting keywords from the query). These query concepts can be searched in the mapping store, to find matching concepts in other natural languages which the user model indicates are also suitable. Once these multilingual concepts are obtained, their associated document instances can be retrieved and returned to the user. To achieve CLIR, a similar process to the above is used in the case study when applying SOCOM to the ARCHING system. This is discussed next.

4.6.3. Design, Implementation and Execution

This section discusses how the matching results generated by SOCOM are consumed by the ARCHING system in the process of achieving CLIR. In particular, this section presents the generation of multilingual ontologies using structured data sources; the generation of CLOM results between them using SOCOM; and the retrieval of documents based on the CLOM results. As mentioned previously, the ARCHING system [Steichen et al., 2011] returns personalised information to a user by adaptively composing and presenting relevant results from structured and unstructured data. Structured data refers to enterprise documentations, and unstructured data refers to user generated content. The retrieval of structured data by ARCHING (using the CLOM results generated by SOCOM) is the focus of this case study. For more information on the retrieval of unstructured data, adaptive composition and presentation that is beyond the scope of this thesis, see [Steichen et al., 2011].

The structured data used in this case study include: real world product manuals (in German) and enterprise technical documentations (in English). These data are provided by Symantec, both describe the home security product: Norton 360. However, they differ in terms of domain coverage and terminology. The German product manual covers a smaller domain, and is written in less technical terms since it is aimed at the general public. The English technical documentation covers a broader domain, and is written for the Symantec employees, hence is more technical. To realise CLOM-enabled CLIR, ontologies were generated by teams of experts (discussed next): an English ontology was generated based on the technical documentations in English and a German ontology was created based on the product manuals in German.

(a) The Technical Documentation Ontology	**(b)** The Product Manual Ontology

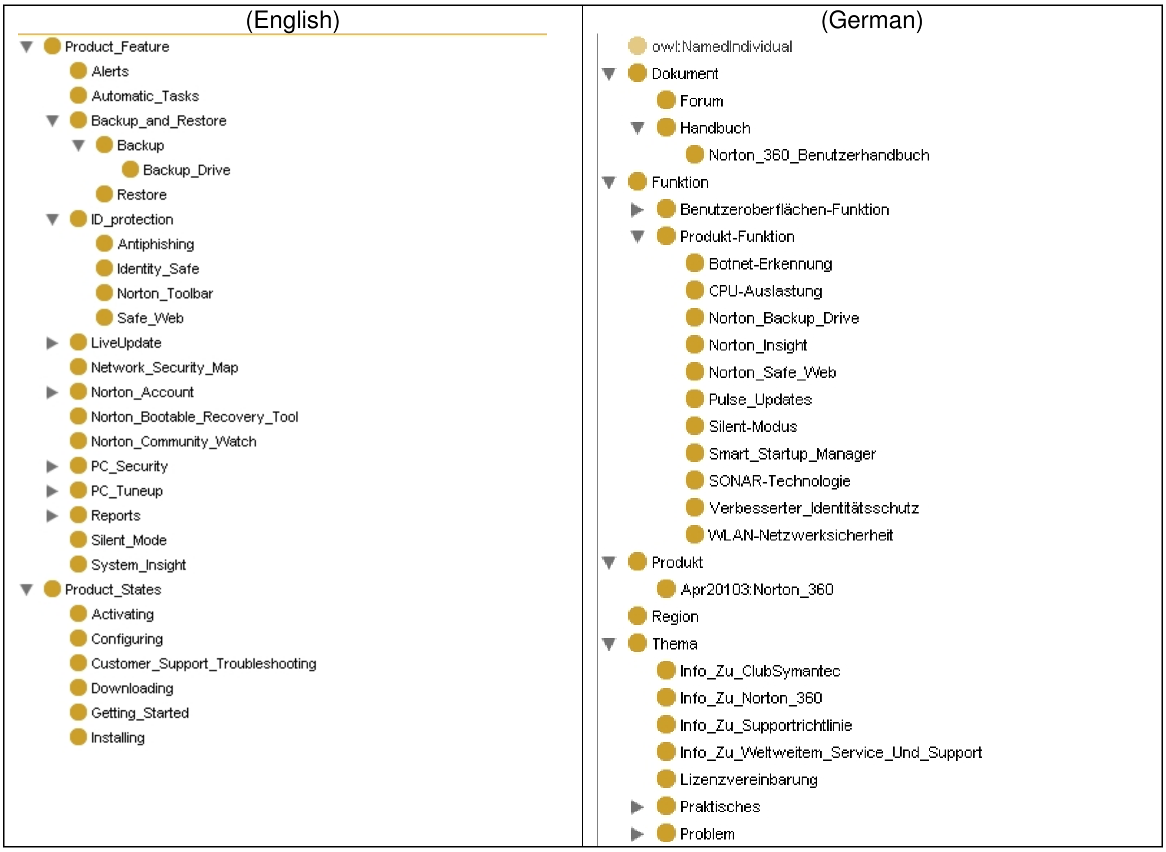


Figure 4-16. The Ontologies in the Case Study

The technical documentations on Norton 360 are available in English and formatted in XML using the DocBook DTD[84], containing elements such as `chapter`, `section`, `paragraph` and `table`. Blocks of text are modelled as instances of paragraph in the ontology. Paragraphs are modelled as sub-classes of sections, and each section is modelled as a child class of a chapter or a child class of another section (i.e. in the case of sections containing sub-sections). These classes are classified under topics of interest. These topics of interest are controlled vocabularies that derived manually by a team of experts that consists of two CNGL members (excluding the author of this thesis). These controlled vocabularies were generated in Protégé by hand using OWL. Finally, the controlled vocabularies were annotated with classes and instances that were automatically extracted from the technical documentation. This extraction process is discussed in [Şah & Wade, 2010]. The final English ontology contains 128 classes, 109 object properties, 36 data type properties and 7523 instances. A partial screenshot of this technical documentation ontology (in English) in Protégé is shown in figure 4-16-a.

The German ontology was generated manually by this author and another Ph.D. candidate who is a native German speaker, using the Protégé editor and the Norton 360 product manuals (PDF files written in English and German). Although the same product

[84] http://www.docbook.org/xml/5.0/dtd/docbook.dtd

manual is available in both English[85] and German[86], as this study aims to demonstrate SOCOM in a CLOM-enabled CLIR system, ontologies in different natural languages are required. As these manuals cover a restricted scope of Norton 360 related topics, it highlights a need to bridge them to the technical documentation ontology that covers a broader scope. There were two steps to the generation of the German ontology: (1) an ontology (in English) was first created manually by this author based on the English product manual; (2) as the German product manual is a direct translation of the English manual (this is confirmed by the native German speaker and a Symantec employee who is collaborating with CNGL), the author-generated ontology is then converted to German by using terminologies from the German version of the same product manual (assisted by the native German speaker). The final product manual ontology in German contains 77 classes, 4 object properties, 4 data type properties and 4 instances. A partial screenshot of this ontology is shown in figure 4-16-b. The ontologies shown in figure 4-16 can be found at `root/SOCOMExperiments/CaseStudy/Ontologies/` on the DVD.

SOCOM was used to generate mappings between the ontologies in English and German. These mappings can be found at `root/SOCOMExperiments/CaseStudy/Mappings/` on the DVD. An overview of how these mappings are used in the ARCHING system is shown in figure 4-17. The German ontology (created from the product manual) and the English ontology (generated from technical documentations where some documents are linked to the entities in the English ontology) are both stored in the eXist database. The mappings between these ontologies are also stored in this database. At runtime, when a query is issued in German, the ARCHING system matches the query to the concept(s) in the German ontology using string comparison techniques. Its matched English concept(s) are then identified by simply searching the mappings that were generated pre-runtime. The identified concept(s) in English is effectively the translation of the original query. The technical contents that are annotated with this English concept(s) are retrieved next. If such readily annotated content is not available (i.e. a mapping does not exist), a text search of the identified English concept(s) in the technical documentation is conducted using the Apache Lucene[87] API. Finally, these contents in English are composed and presented to the user.

[85] ftp://ftp.symantec.com/public/english_us_canada/products/norton_360/
[86] ftp://ftp.symantec.com/public/deutsch/produkten/norton_360/
[87] http://lucene.apache.org/java/docs/index.html

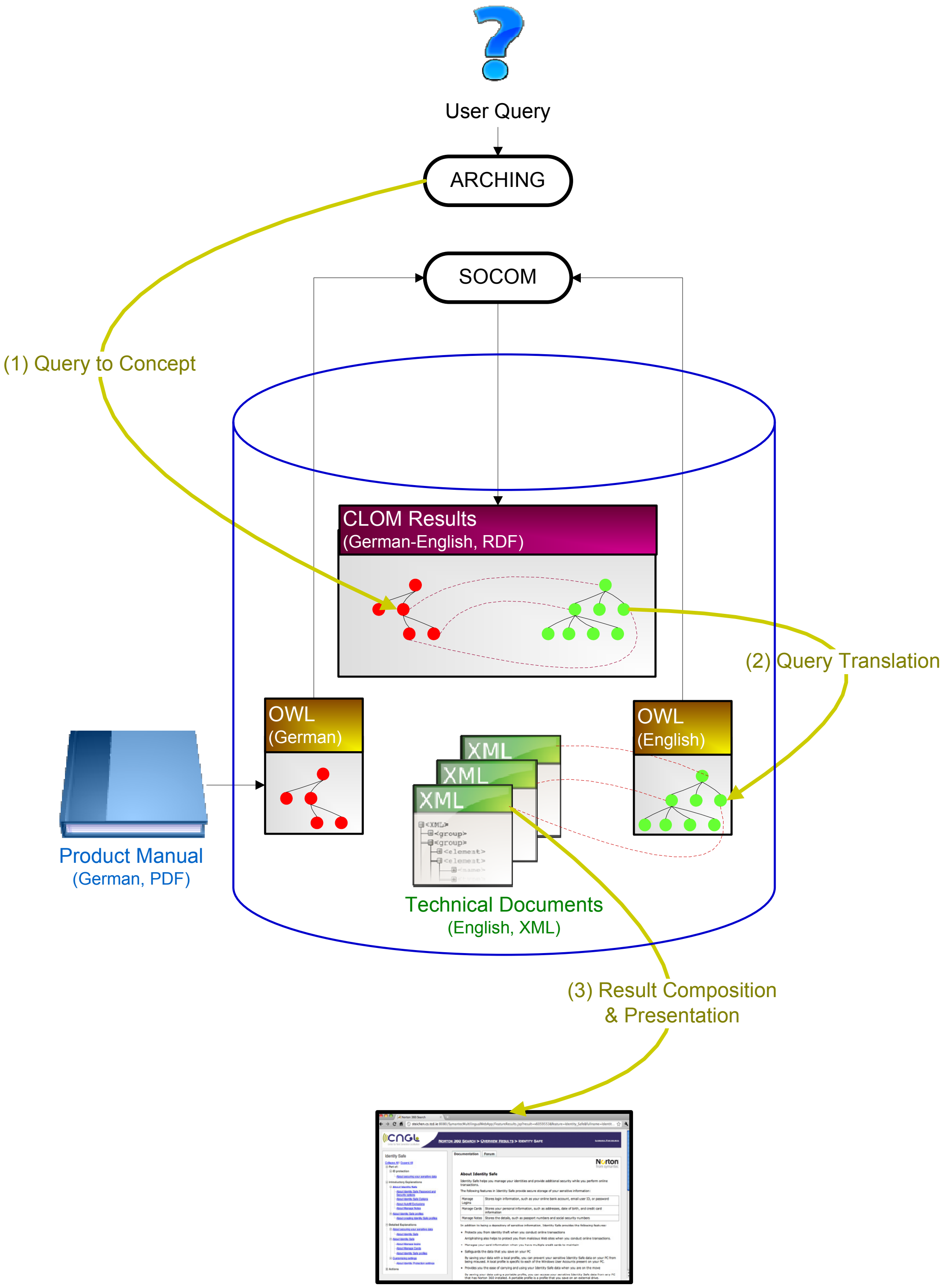
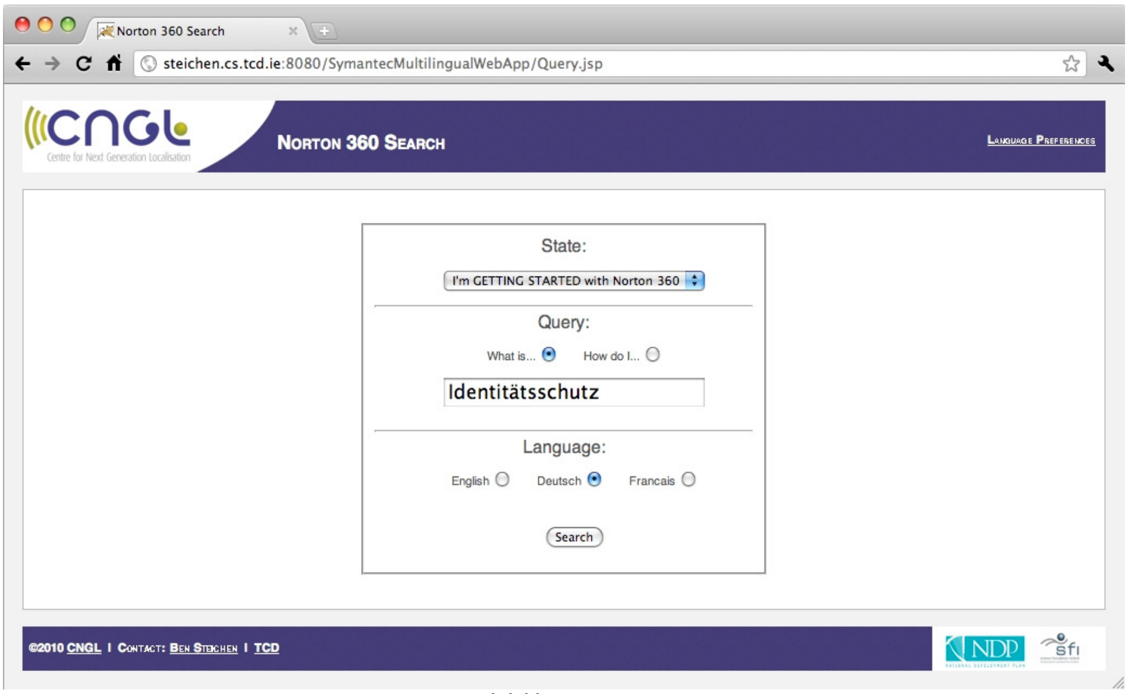


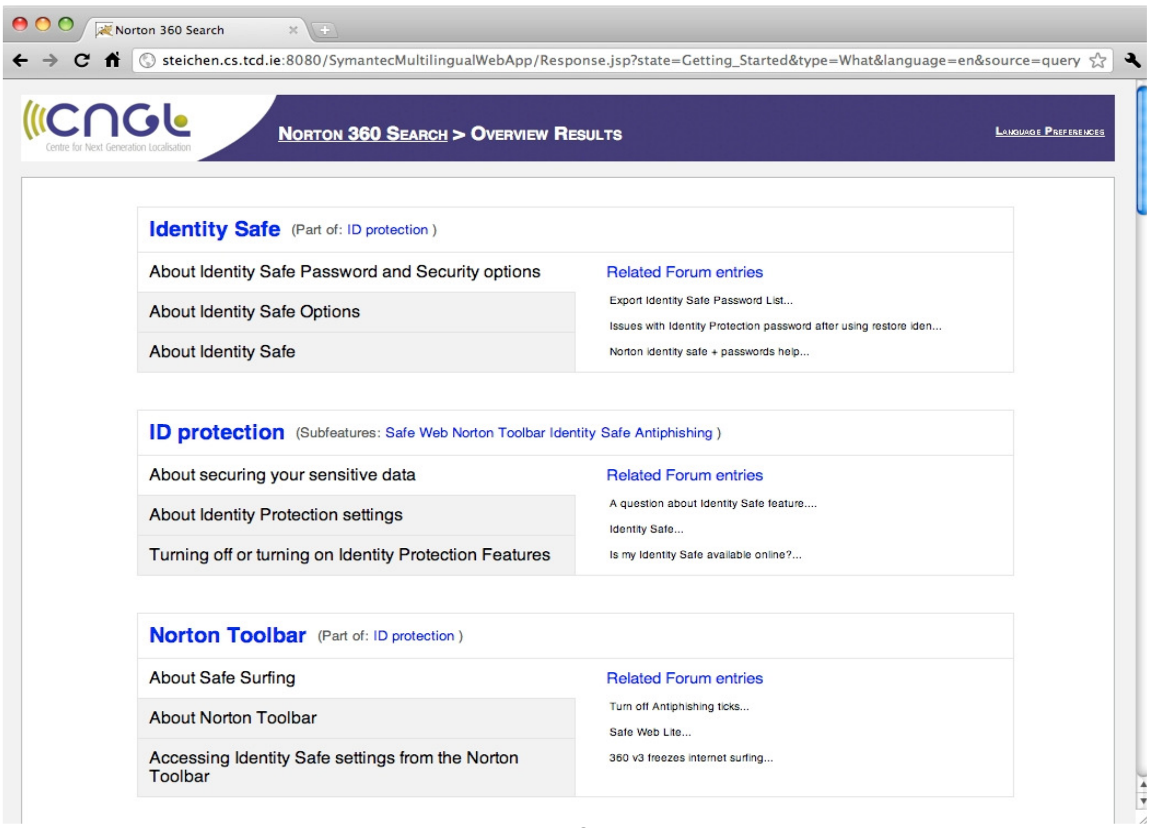
Figure 4-17. SOCOM in CLIR

Figure 4-18 shows the screenshots from the ARCHING system that retrieves documents using the CLOM results generated by SOCOM. Figure 4-18-a shows the homepage of the system, where a user specifies the attributes in the user model: state (e.g. getting started with Norton 360, or reacting to a problem), query (i.e. the search query), query intent (is the query a *what* question or a *how* question, where the former focuses on explanations and the latter focuses on instructions) and language (e.g.

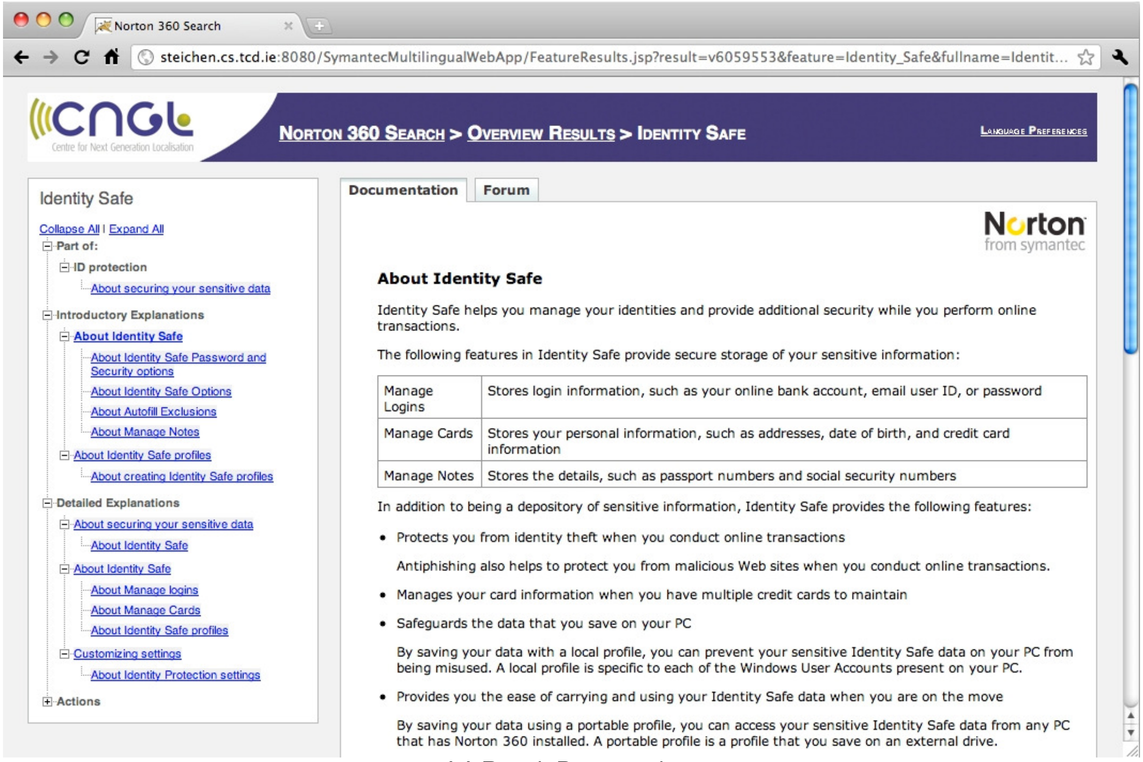
German). These attributes influence the order of the grouped results shown in figure 4-18-b. When a group is selected by the user, the most relevant technical content (in English) is displayed with its branches in the ontology already expanded in the navigation list (see figure 4-18-c). For more information on how groups are constructed, ranked and how they are adapted to the user model which is outside the scope of this thesis, see [Steichen & Wade, 2010].



(a) Homepage



(b) Result Grouping



(c) Result Presentation

Figure 4-18. Screenshots from the ARCHING System

4.6.4. Significance of the Case Study

This section presents the objectives met and the significance of the case study.

Objective (1): *demonstrate the feasibility of SOCOM in a real world application: ARCHING* is met through the successful application of CLOM results (generated by SOCOM) in the process of achieving cross-lingual document retrieval (through the ARCHING system). The study shows the potential of the SOCOM system to solve a real world problem. The strategy undertaken in the study (i.e. using CLOM techniques to achieve CLIR) is a novel approach to achieve cross-lingual information retrieval. The study leverages relevant resources that are available in different natural languages to the multilingual user using CLOM results, and serves as a proof of concept for the application of the SOCOM in CLIR systems.

Objective (2): *investigate the potential benefits and drawbacks from using CLOM techniques for the purpose of CLIR*. The ontologies involved in the case study concern new domains and natural language pairs (in addition to what was shown in section 4.5), which successfully shows the SOCOM's ability to work with ontologies outside the laboratory experiments discussed in section 4.5. The generation of the multilingual ontologies (in English and German) took four weeks collectively, as it required discussions (in the case of the German ontology) and systems to process the structured technical documentations (in the case of the English ontology). The generation of the CLOM results (between the English and the German ontology) took two working days as the English ontology was relatively large. Once these CLOM results are stored in the database, the retrieval at runtime took seconds. The benefits of using CLOM techniques for the purpose of CLIR can be summarised as:

- SOCOM is able to work with ontologies and natural languages outside the laboratory experiments shown (in section 4.5);
- the effort required to generate the CLOM results using SOCOM is relatively small (e.g. a small configuration in SOCOM was necessary for the MT tools so that the source natural language is set to German, and the target natural language is set to English).

However, there are some drawbacks to applying CLOM techniques to CLIR, including:

- ontology construction overhead can be time-consuming (as seen in the case study, the generation of the German and the English ontology took the most time);
- errors (i.e. incorrect mappings) can occur in the CLOM results generated by SOCOM, which can lead to poor documents presented to the user. For example, *PC_Optimierung* (from the product manual ontology, meaning "personal computer optimisation" in German) was matched to *Disk_Optimization* (from the technical documentation ontology in English) when using the *SubsDistNameAlignment* algorithm (see the file located at `root/SOCOMExperiments/CaseStudy/Mapping/SubsDistName.rdf` on the DVD), which led the system to incorrectly retrieve documents related to disk optimisation instead of PC optimisation.
- not all entities in the German ontology are mapped to the entities in the English ontology since the two contain overlapping concepts. When mappings simply do not exist, the system fails to associate German queries with English concepts and needs to rely on text search on the web.

Though the case study shows some drawbacks of applying CLOM techniques in CLIR, however, the significance of the study is undiminished. Using CLOM results is a novel approach to overcome natural language barriers in IR systems. This study validates the soundness of the approach while presenting alternative avenues for future research in cross-lingual information retrieval.

4.7. Summary

Based on the findings presented in chapter 3, this chapter presents the appropriate ontology label translation (AOLT) concept in the context of CLOM. In addition, an initial prototype: the Semantic-Oriented Cross-lingual Ontology Mapping (SOCOM) system is designed and developed to assist the AOLT selection process in an effort to improve CLOM quality. SOCOM integrates a basic implementation of the AOLT component that makes use of a minimum set of ontological semantics in order to select AOLT results in the process of generating CLOM results. This prototype serves as a proof of concept for the use of the AOLT process in CLOM. The goal of SOCOM is to apply appropriately selected translations in order to improve mapping quality. The evaluation of SOCOM thus focuses on the validation of the AOLT concept, where findings shown an improvement in the mapping quality given the basic AOLT process. In addition, a case study is presented in this chapter, where SOCOM is applied to an adaptive information retrieval, composition and presentation system named ARCHING. This study serves as a proof of concept for CLOM-enabled CLIR systems.

5 PROTOTYPE TWO: SOCOM++

5.1. Chapter Overview

A basic AOLT process (with minimum intake of the source and target ontology semantics that are always available to a mapping scenario) has shown to be effective at improving CLOM quality (discussed in chapter 4). This finding naturally motivates further research on whether an improved AOLT process - for instance, one that accounts more inputs than the basic AOLT process in SOCOM - could gain further improvement on the CLOM quality. It is now known that the mapping outcome differs depending on the translations of the ontology labels (since the only difference between SOCOM and the baseline system is the translations used for the source ontology during ontology rendition). It is thus of interest to investigate whether support can be provided for adjusting the AOLT outcome in a given CLOM scenario, through for example, the use of configurable inputs of the AOLT process. This chapter presents the second prototype: SOCOM++, which aims to address the above issues.

SOCOM++ incorporates a more sophisticated AOLT component which allows adjustment on the AOLT outcome in an effort to influence the final mapping outcome. Improved from SOCOM, SOCOM++ offers additional inputs to the AOLT component, which are also configurable for the mapping expert. Such a design aims to facilitate the tuning of SOCOM++ in specific cross-lingual ontology mapping environments. The evaluation of SOCOM++ focuses on the adjustment of the mapping quality given the same ontology pair, and aims to demonstrate evidently that the mapping outcome can be adjusted in the same cross-lingual ontology mapping scenario when different AOLT results are selected. The evaluation of SOCOM++ uses the same two pairs of ontologies that were previously used in the evaluation of SOCOM. In order to investigate the impact of different AOLT settings on the quality of the mappings generated, a total of six experimental trials have been conducted where each trial focuses on one aspect of the configurable features. In addition, scalability tests have been carried out to investigate how two different trial configurations cope with increased workload. Note that the trials presented in this chapter are not an exhaustive list of all possible configurations of SOCOM++, but rather examples of typical adjustments that can be made to the AOLT selection process.

The evaluation results shown in section 5.4.3.3. (experiment two) have been published in the paper titled *Using Pseudo Feedback to Improve Cross-Lingual Ontology Mapping*, at the 8^{th} Extended Semantic Web Conference (ESWC 2011), LNCS 6643, pp. 336-351, in May 2011. The remainder of this chapter is organised as follows. Section 5.2 presents the design of SOCOM++. This is followed by the implementation details in section 5.3. The evaluation of SOCOM++ is presented in section 5.4. Finally, section 5.5 concludes this chapter with a summary.

5.2. SOCOM++ Design

This section presents an overview of the design of SOCOM++. The processes involved are outlined in figure 5-1. As discussed previously (in chapter 4, section 4.2), there are other ways to achieve AOLT results in the context of CLOM such as expert-based or rule-based, the AOLT process shown in SOCOM++ and SOCOM are examples of how AOLT results can be achieved.

The core steps to achieve cross-lingual ontology mapping in SOCOM++ are generally similar to what is seen in SOCOM, in that O_1 is transformed to O_1' via the *ontology rendition* process, which is then matched to O_2 via the *MOM* process. However in SOCOM++, when choosing the AOLT results for labels in O_1, the *AOLT selection* process accounts several additional inputs compared to SOCOM. Besides analysing the O_1 semantics and O_2 semantics (similarly to SOCOM), SOCOM++ also accounts four other inputs in the process of generating AOLT results, including execution constraint, resource constraint, task intent and feedback as shown in figure 5-1. All six inputs can be configured by the user - their configuration details are discussed later (in section 5.3). An overview of each input is presented here.

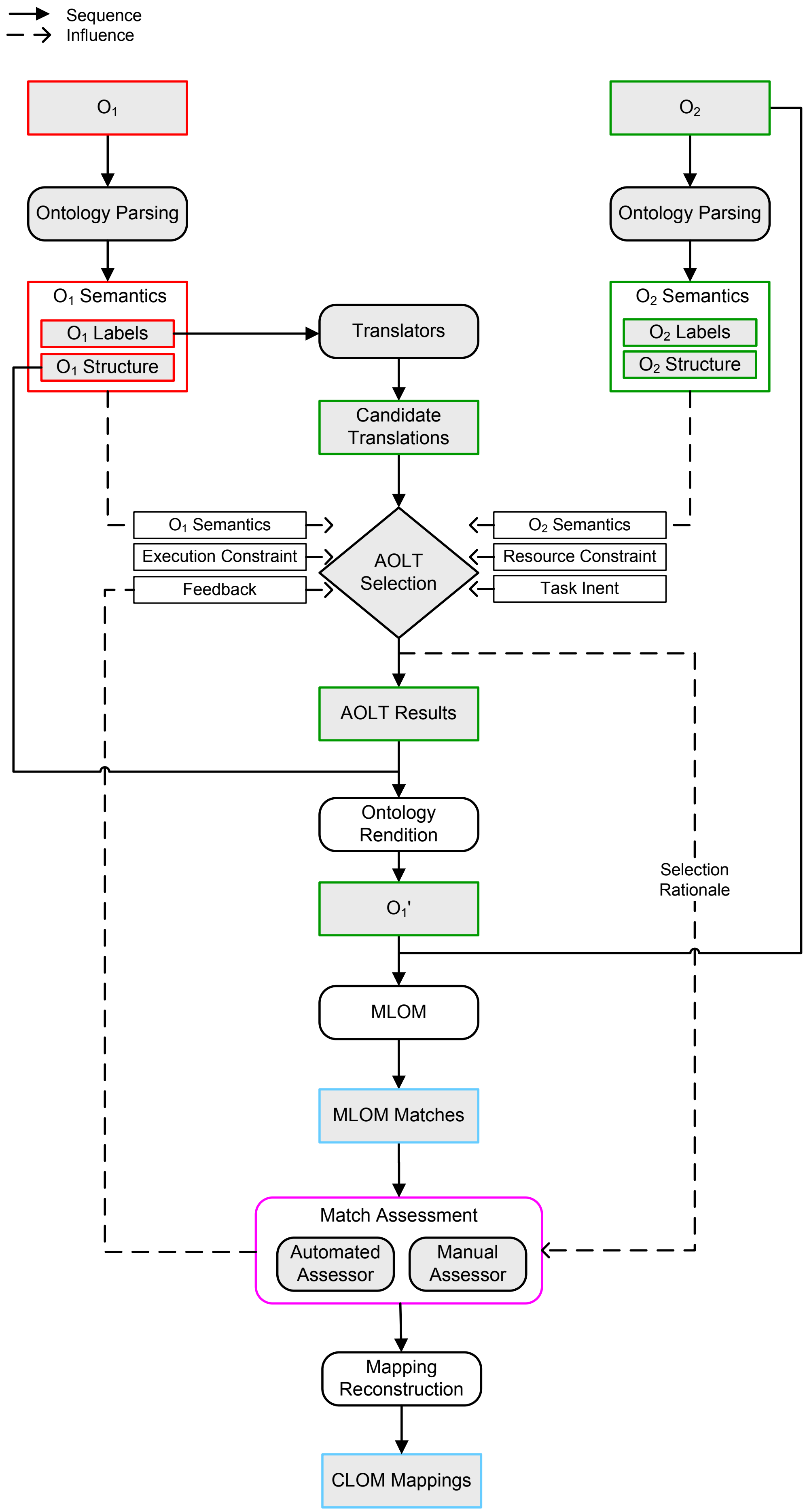


Figure 5-1. Prototype Two: SOCOM++ Design Diagram

- *O_1 semantics* refer to the embedded and background semantics of ontological entities in a given source ontology. Embedded semantics refer to formally defined resources in a given ontology such as the semantic surroundings of entities. Background semantics refer to knowledge drawn from external resources such as dictionaries and thesauri. In SOCOM++, synonyms of the candidate translations for O_1 labels are collected which differs from the design of prototype one (recall from chapter 4, section 4.3, figure 4-2, the translation repository only contained candidate translations of labels in O_1). This increases the size of the selection pool for the AOLT results, which presents the system with more candidate AOLT results for a given label. Also, additional alternative translations will be available to the system when a collision is encountered.
- *O_2 semantics* refer to the embedded and background semantics of ontological entities in a given target ontology. Similarly to the O_1 semantics discussed above, embedded semantics are formally defined and background semantics are externally concluded. More precisely, the embedded semantics in O_2 include the labels used by target entities and their semantic surroundings. The background semantics in O_2 include the synonyms generated for the target labels.
- *Execution constraint* is a high-level restriction on how the AOLT selection process will be run. It offers the user with a choice of performing the default system configuration without having to specify values for any other configurable input. By having a default configuration, the user can generate initial mappings in a CLOM scenario, analyse the mapping outcome and decide on the specific adjustment on the AOLT process for further tuning of the mapping outcome.
- *Resource constraint* is the availability of external resources (e.g. dictionaries, thesaurus) that are available to the AOLT selection process. In SOCOM++, this includes the availability of synonyms in the given ontology domain. A lack of synonyms may be evident in some specialised domains (e.g. medical) whereby there are few other ways to express the same concept, or synonyms are simply not available/accessible.
- *Task intent* is a representation of the motivation for the mapping activity being carried out. For example, the intent can be to increase mapping

precision (i.e. generating as many correct matches as possible), or to increase mapping recall (i.e. generating as many matches as possible to ensure the completeness of the mappings).

- *Feedback* aims to improve the matching quality upon recognising how correct matches have been achieved. By assessing the candidate matches generated in a specific CLOM scenario via *automated assessor* (e.g. infer the correctness of the matches without the involvement of a user using pseudo feedback) or *manual assessor* (e.g. explicit feedback from a user), the system attempts to improve its future selection of the translations based on the AOLT *selection rationale* derived from this assessment process. The feedback feature in SOCOM++ is inspired by the relevance feedback mechanism that is commonly used in the field of IR.

Ruthven & Lalmas [Ruthven & Lalmas, 2003] present an extensive survey on relevance feedback used in IR. Broadly speaking, there are three types of relevance feedback: explicit, implicit and pseudo feedback. Explicit feedback is obtained after a query is issued by the user and an initial set of documents is retrieved, the user marks these initial documents as relevant or not relevant, and the system retrieves a better list of documents based on this feedback by computing a single or multiple iterations. Implicit feedback works similarly but attempts to infer users' intentions based on observable behaviour. Pseudo feedback is generated when the system makes assumptions on the relevancy of the retrieved documents. In the context of ontology mapping, the use of explicit user feedback is successfully demonstrated in monolingual ontology mapping [Duan et al., 2010]. SOCOM++ expands on Duan et al.'s work and applies a pseudo feedback technique (i.e. without the involvement of a user) in CLOM scenarios. Assumptions on matches' correctness are based on their confidence levels in SOCOM++. There are many ways to calculate confidence levels as documented by Euzenat & Shvaiko [Euzenat & Shvaiko, 2007]. Although currently there is no obvious method that is a clear success [Ichise, 2009], confidence levels nonetheless are a way to perceive the probability of a match being correct or not. In SOCOM++, the feedback feature assumes after an initial execution that matches with confidence levels above a certain threshold are correct. It then examines how these matches are generated. Currently, this involves examining which translation media were used (i.e. selection rationale). The rationale then influences the selection of AOLT results in the second iteration of the AOLT process.

To initiate SOCOM++, a user is required to configure the inputs discussed above. The users of SOCOM++ are anticipated to have knowledge in ontologies and ontology mapping. Typical use of the system will likely to include several executions of SOCOM++, whereby a user will determine the configurations for the first run of the system, examine the matches generated from this initial run and adjust variable settings for the second run of SOCOM++. This is repeated until the user achieves the desired mappings or terminates the system when improvement is no longer evident in the mappings. The implementation details of SOCOM++ are discussed next.

5.3. SOCOM++ Implementation

The tools and technologies used in SOCOM++ are presented and discussed in this section. The source code of SOCOM++ can be found at `root/SOCOM++/` on the DVD.

System configuration: a configuration file using the Java utility class `Properties`[88] representing a persistent set of properties formatted in XML (see figure 5-2) and is read at the start-up of SOCOM++ to instruct the execution of the AOLT process. This configuration file contains the variable values that have been set by the user. It follows the DTD (see appendix D, section D.3, figure D-3) defined by Sun Microsystems[89]. How the AOLT inputs are modelled in SOCOM++ is discussed next.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
	<!--execution constraint-->
	<entry key="default">true</entry>
	<!--O1 semantics-->
	<entry key="sourceSurrounding">false</entry>
	<!--O2 semantics-->
	<entry key="targetSurrounding">false</entry>
	<!--resource constraint-->
	<entry key="translationSynonym">true</entry>
	<entry key="targetSynonym">true</entry>
	<!--task intent-->
	<entry key="correctnessOptimise">false</entry>
	<entry key="completenessOptimise">false</entry>
	<!--pseudo Feedback-->
	<entry key="threshold">0.5</entry>
</properties>
```

Figure 5-2. The Configuration File in SOCOM++

Execution constraint is modelled by the *entry* element with the *key* attribute: `default` in figure 5-2. It can be configured to either `true` or `false`, whereby `true`

[88] http://download.oracle.com/javase/1.5.0/docs/api/java/util/Properties.html
[89] Renamed Oracle America, Inc. in 2010.

initiates SOCOM++ to run a default setting of the AOLT selection process, and `false` instructs the AOLT process to run according to the values set for the other *entry* elements. When it is set to `true`, the values configured for all other *entry* elements are ignored by the prototype. In other words, execution constraint offers the user the choice between an automated execution of the prototype or a tailored execution with desired configurations of other *entry* elements depending on the specific mapping scenario.

O_1 semantics is modelled by the *entry* element with the *key* attribute: `sourceSurrounding` in figure 5-2. It can be configured to either `true` or `false`, with the former instructing the AOLT process to take the semantic surroundings of source entities into account, and the latter instructing the AOLT process to not consider the semantic surroundings of the source entities during the selection process. In addition to generating candidate translations for labels in O_1 (as seen in SOCOM), SOCOM++ also generates synonyms for these candidate translations.

Similarly, *O_2 semantics* is modelled by the *entry* element with the *key* attribute: `targetSurrounding` in figure 5-2, which can be set to either `true` or `false`. A value `true` allows the AOLT selection process of take the semantic surroundings of the target entities into account, and a value `false` instructs the AOLT selection process to disregard the semantic surroundings of the target entities. Synonyms for labels in O_2 are generated in SOCOM++ as was done in SOCOM.

Resource constraint is modelled by the *entry* element with the *key* attribute: `translationSynonym` and the *entry* element with the *key* attribute: `targetSynonym`. Both elements can be configured to either `true` or `false`, and are designed to offer the user the option to restrict external resources during an AOLT selection process. If the `translationSynonym` is set to `true`, the synonyms generated for candidate translations of the source labels are accounted during the AOLT selection. If it is set to `false`, the AOLT process will not consider these synonyms. Similarly, if the `targetSynonym` is set to `true`, the AOLT process will include the synonyms collected for the O_2 labels during the selection process. If it is set to `false`, the opposite will occur.

Task intent is modelled by the *entry* element with the *key* attribute: `correctnessOptimise` and the *entry* element with the *key* attribute: `completenessOptimise`. Both can be configured to either `true` (i.e. enabling a feature) or `false` (i.e. disabling a feature), however, only one of these elements can be set to `true` at a time. This is because the current implementation can only aim to improve either just the correctness or just the completeness of the matches generated, but not both at the same time. Optimising correctness is achieved by assuming only the matches generated from the first iteration with 1.0 confidence levels are correct, analysing how they were achieved (in SOCOM++, this involves identifying the MT tools used to generate these correct matches) and compute a second iteration of the AOLT process. Optimising completeness is achieved by assuming all matches (i.e. with any confidence level) generated from the first iteration are correct, analysing how they were achieved and computing a second iteration of the AOLT selection process accordingly. Correctness is optimised by strictly eliminating uncertain matches (i.e. any match that has less than 1.0 confidence level), and attempts to increase the number of certain matches (i.e. matches with 1.0 confidence levels) which in turn optimises mapping precision. During this process, it is possible that correct matches are eliminated (i.e. those matches that have lower than 1.0 confidence levels, but are still correct). Hence in contrast, completeness optimisation avoids incorrect eliminations of uncertain matches (since all matches in the first iteration are assumed to be correct), which is a much more relaxed strategy (in comparison to optimising correctness) to increase correct matches.

SOCOM++ integrates a *pseudo feedback* feature, which is modelled by the *entry* element with the *key* attribute: `threshold` in figure 5-2. Its value can be set to anything between `0.0` and `1.0`. The threshold is a cut-off point for the confidence levels that enables the pseudo feedback to speculate which matches generated may be correct. For example, when the threshold is set to 0.75, the pseudo feedback feature assumes those matches with at least 0.75 confidence levels are correct. This feature can be considered as being a middle ground between two extremes - one extreme being the optimisation of the completeness and the other being the optimisation of the correctness (as modelled in the task intent feature). The task intent and the pseudo feedback feature are different options for SOCOM++ to carry out a second iteration of the AOLT process, which means only one task intent (either optimising correctness or optimising completeness) or pseudo feedback can be in effect at a time.

AOLT Selection in SOCOM++: To facilitate the selection of AOLT results given the aforementioned configurable inputs, SOCOM++ carries out three main steps including *semantic analysis*, *ontology rendition* and *ontology mapping* to achieve cross-lingual ontology mapping as shown in figure 5-3. Each step is discussed next.

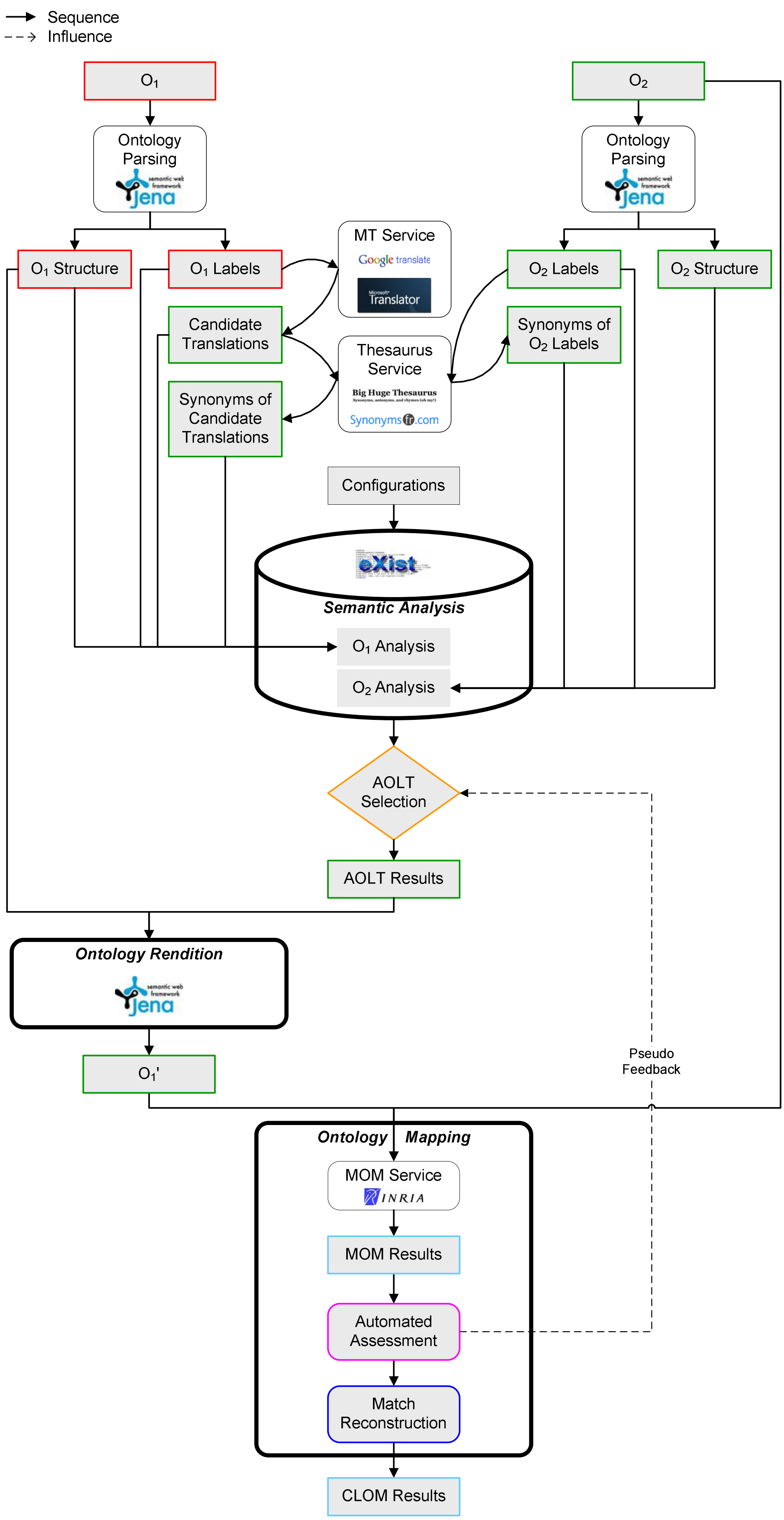


Figure 5-3. Implementation Diagram of SOCOM++

The Semantic Analysis Step: the Jena framework 2.5.7 is used to parse the given ontologies, extracts the resource labels and their corresponding semantic surroundings. To generate candidate translations for ontology labels in O_1, the GoogleTranslate API 0.5 and the Microsoft Translator API[90] are integrated by the *MT Service* shown in figure 5-3. In addition, synonyms are also generated for these candidate translations via the *Thesaurus Service*. In SOCOM++, the *Thesaurus Service* uses the Big Huge Thesaurus API[91] for synonyms in English and the synonyms-fr.com website for synonyms in French. The code snippet in appendix C, section C.4, figure C-13 demonstrates how synonyms are generated via the Big Huge Thesaurus API. The code snippet in appendix C, section C.4, figure C-14 illustrates how synonyms are generated via the synonyms-fr.com. The outcomes from processing the source ontology, including the original O_1 labels, their semantic surroundings, their candidate translations and the corresponding synonyms of these candidate translations are formatted in XML and stored in the eXist DB version 1.4, as *O_1 Analysis* shown in figure 5-3. Similarly, the outcomes from processing the target ontology, including the original labels in O_2, their semantic surroundings and corresponding synonyms are also formatted in XML and stored in the eXist database version 1.4, as *O_2 Analysis* shown in figure 5-3. An example of *O_1 analysis* is shown in figure 5-4 and an example of *O_2 analysis* is shown in figure 5-5. The DTD declared for O_1 analysis can be found in appendix D, section D.3, figure D-4. The DTD declared for O_2 analysis can be found in appendix D, section D.3, figure D-5.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SourceSemantic SYSTEM "SourceSemantic.dtd">
<SourceSemantic>
  …
  <Resource id="CLS11">
    <OntLabel>学术会议</OntLabel>
    <MTLabel>学术会议</MTLabel>
    <Translation>
      <Candidate id="CDD0-CLS11">
        <CandidateValue>Conference</CandidateValue>
        <CandidateSource>google</CandidateSource>
        <CandidateConcatenated>Conference</CandidateConcatenated>
        <CandidateSynonymCollection>
          <CandidateSynonym      concatenated="discussion"      id="SYN0-CDD0-CLS11"
          source="BHT" value="discussion"/>
          <CandidateSynonym     concatenated="group_meeting"     id="SYN1-CDD0-CLS11"
          source="BHT" value="group meeting"/>
          <CandidateSynonym concatenated="league" id="SYN2-CDD0-CLS11" source="BHT"
          value="league"/>
          <CandidateSynonym      concatenated="association"      id="SYN3-CDD0-CLS11"
          source="BHT" value="association"/>
        </CandidateSynonymCollection>
      </Candidate>
      <Candidate id="CDD1-CLS11">
        <CandidateValue>Academic conferences</CandidateValue>
        <CandidateSource>bing</CandidateSource>
        <CandidateConcatenated>Academic_conferences</CandidateConcatenated>
      </Candidate>
    </Translation>
```

[90] http://www.microsofttranslator.com/dev
[91] http://words.bighugelabs.com/api.php, its data is based on the Princeton University WordNet database, the Carnegie Mellon Pronouncing Dictionary, and crowd-sourced suggestions.

```
    <Surrounding MTLabel="事件" OntLabel="事件" id="CLS48"/>
  </Resource>
  …
  <Resource id="CLS48">
    <OntLabel>事件</OntLabel>
    <MTLabel>事件</MTLabel>
    <Translation>
      <Candidate id="CDD0-CLS48">
        <CandidateValue>Event</CandidateValue>
        <CandidateSource>google</CandidateSource>
        <CandidateConcatenated>Event</CandidateConcatenated>
        <CandidateSynonymCollection>
          <CandidateSynonym concatenated="case" id="SYN0-CDD0-CLS48" source="BHT"
          value="case"/>
          <CandidateSynonym concatenated="consequence" id="SYN1-CDD0-CLS48"
          source="BHT" value="consequence"/>
          <CandidateSynonym concatenated="effect" id="SYN2-CDD0-CLS48" source="BHT"
          value="effect"/>
          <CandidateSynonym concatenated="outcome" id="SYN3-CDD0-CLS48" source="BHT"
          value="outcome"/>
          <CandidateSynonym concatenated="result" id="SYN4-CDD0-CLS48" source="BHT"
          value="result"/>
          <CandidateSynonym concatenated="issue" id="SYN5-CDD0-CLS48" source="BHT"
          value="issue"/>
          <CandidateSynonym concatenated="circumstance" id="SYN6-CDD0-CLS48"
          source="BHT" value="circumstance"/>
          <CandidateSynonym concatenated="phenomenon" id="SYN7-CDD0-CLS48"
          source="BHT" value="phenomenon"/>
        </CandidateSynonymCollection>
      </Candidate>
      <Candidate id="CDD1-CLS48">
        <CandidateValue>Event</CandidateValue>
        <CandidateSource>bing</CandidateSource>
        <CandidateConcatenated>Event</CandidateConcatenated>
        <CandidateSynonymCollection>
          <CandidateSynonym concatenated="case" id="SYN0-CDD1-CLS48" source="BHT"
          value="case"/>
          <CandidateSynonym concatenated="consequence" id="SYN1-CDD1-CLS48"
          source="BHT" value="consequence"/>
          <CandidateSynonym concatenated="effect" id="SYN2-CDD1-CLS48" source="BHT"
          value="effect"/>
          <CandidateSynonym concatenated="outcome" id="SYN3-CDD1-CLS48" source="BHT"
          value="outcome"/>
          <CandidateSynonym concatenated="result" id="SYN4-CDD1-CLS48" source="BHT"
          value="result"/>
          <CandidateSynonym concatenated="issue" id="SYN5-CDD1-CLS48" source="BHT"
          value="issue"/>
          <CandidateSynonym concatenated="circumstance" id="SYN6-CDD1-CLS48"
          source="BHT" value="circumstance"/>
          <CandidateSynonym concatenated="phenomenon" id="SYN7-CDD1-CLS48"
          source="BHT" value="phenomenon"/>
        </CandidateSynonymCollection>
      </Candidate>
    </Translation>
    <Surrounding MTLabel="展览" OntLabel="展览" id="CLS32"/>
    <Surrounding MTLabel="课程" OntLabel="课程" id="CLS40"/>
    <Surrounding MTLabel="研讨会" OntLabel="研讨会" id="CLS46"/>
    <Surrounding MTLabel="会议" OntLabel="会议" id="CLS5"/>
    <Surrounding MTLabel="学术会议" OntLabel="学术会议" id="CLS11"/>
  </Resource>
  …
</SourceSemantic>
```

Figure 5-4. An Example of O_1 Analysis

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TargetSemantic SYSTEM "TargetSemantic.dtd">
<TargetSemantic>
  …
  <Resource id="CLS3">
    <OntLabel>Researcher</OntLabel>
    <MTLabel>Researcher</MTLabel>
    <SynonymCollection>
      <Synonym concatenated="research_worker" id="SYN0-CLS3" source="BHT"
      value="research worker"/>
      <Synonym concatenated="investigator" id="SYN1-CLS3" source="BHT"
      value="investigator"/>
      <Synonym concatenated="scientist" id="SYN2-CLS3" source="BHT"
      value="scientist"/>
    </SynonymCollection>
    <Surrounding MTLabel="Person" OntLabel="Person" id="CLS16"/>
  </Resource>
```

```
…
  <Resource id="CLS16">
    <OntLabel>Person</OntLabel>
    <MTLabel>Person</MTLabel>
    <SynonymCollection>
      <Synonym concatenated="individual" id="SYN0-CLS16" source="BHT"
      value="individual"/>
      <Synonym concatenated="someone" id="SYN1-CLS16" source="BHT"
      value="someone"/>
      <Synonym concatenated="somebody" id="SYN2-CLS16" source="BHT"
      value="somebody"/>
      <Synonym concatenated="being" id="SYN3-CLS16" source="BHT" value="being"/>
      <Synonym concatenated="cause" id="SYN4-CLS16" source="BHT" value="cause"/>
      <Synonym concatenated="figure" id="SYN5-CLS16" source="BHT" value="figure"/>
    </SynonymCollection>
    <Surrounding MTLabel="Employee" OntLabel="Employee" id="CLS18"/>
    <Surrounding MTLabel="Researcher" OntLabel="Researcher" id="CLS3"/>
    <Surrounding MTLabel="Student" OntLabel="Student" id="CLS2"/>
    <Surrounding MTLabel="Faculty Member" OntLabel="Faculty_Member" id="CLS28"/>
  </Resource>
  …
</TargetSemantic>
```

Figure 5-5. An Example of O_2 Analysis

A UML class diagram illustrating the semantic analysis process (that generates output such as the examples shown in figure 5-4 and figure 5-5) is presented in figure 5-6. The *OntologyParser* class is responsible for loading a given ontology and creating an `OntModel`[92] for it via the Jena framework for further semantic processing (i.e. extract resource labels, generate semantic surroundings for a given resource). For a source ontology, the *SemanticProcessor* class then initiates the *SourceUpdater* class to extract and store the embedded semantics for ontological classes, object properties, data type properties and individuals. For a target ontology, this is achieved by the *TargetUpdater* class. A unique ID is assigned to each `Resource` element (which can be an ontological class, an object property, a data type property or an individual) as the value of the attribute `id` as shown in figure 5-4 and figure 5-5. The original resource label is stored as the content of the `OntLabel` element. To break up concatenated labels, the *LabelProcessor* class is called, and the label in natural language form is stored as the content of the `MTLabel` element. The *SurroundingGenerator* class is responsible for generating semantic surroundings for a given ontological resource, which are stored as the `Surrounding` element with the attribute `id` (which is a reference identifier), `OntLabel` and `MTLabel` in figure 5-4 and figure 5-5. For a source ontology, candidate translations with unique IDs are collected via the *MTService* class and stored in the `Candidate` element with child elements `CandidateValue` (the translation returned from a MT tool), `CandidateSource` (the MT tool which returned this translation) and `CandidateConcatenated` (the translation with removed white spaces) as shown in figure 5-4. To collect synonyms for the candidate translations, the *ThesaurusService* class is called. These synonyms are stored in the `CandidateSynonym` element under the

[92] OntModel is an interface from the Jena framework that wraps the underlying model of a given ontology.

parent element `CandiateSynonymCollection`, with attributes `id` (unique ID for a synonym), `source` (the thesaurus used to generate this synonym), `value` (the synonym in natural language form) and `concatenated` (the synonym without any white space). Similarly, the *TargetUpdater* class calls the *ThesaurusService* class to generate synonyms for the target resource labels, which are stored in the `Synonym` element under the parent element `SynonymCollection`, with attributes `id` (the unique ID for this synonym), `source` (the thesaurus that returned this synonym), `value` (the synonym in natural language form) and `concatenated` (the synonym with removed white spaces).

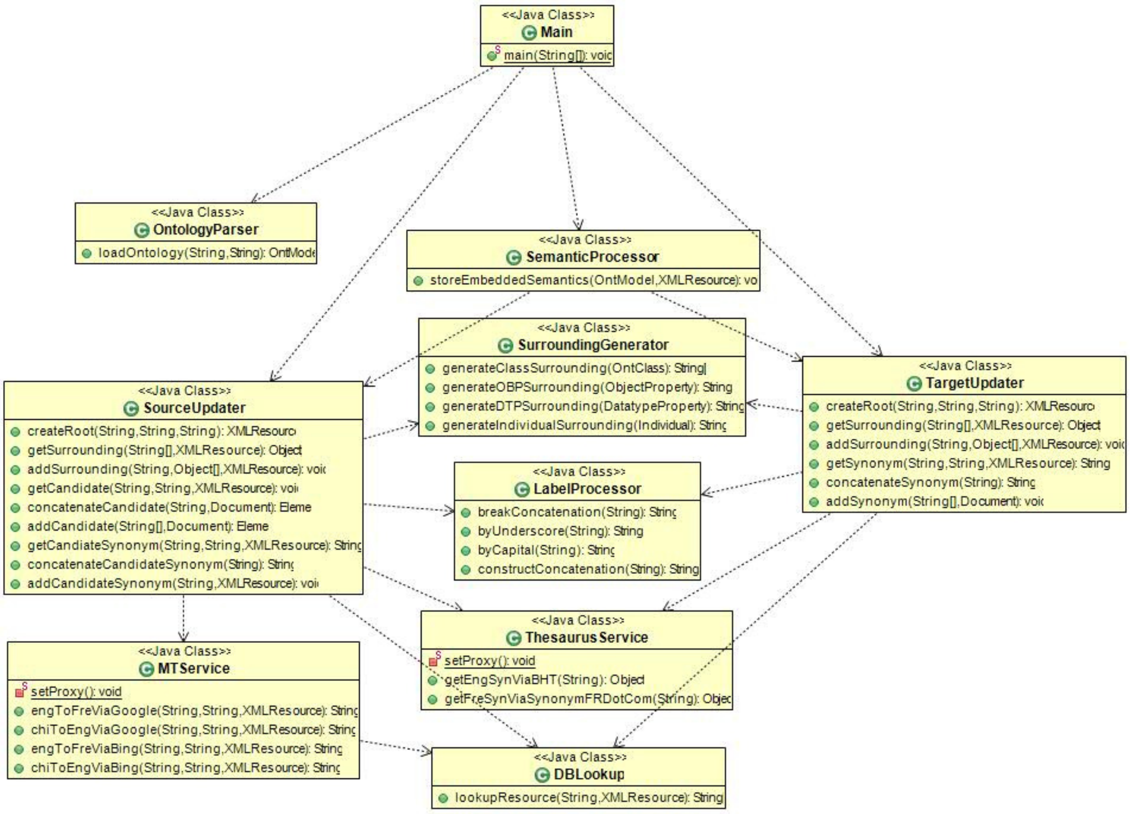


Figure 5-6. Class Diagram of Ontology Semantic Analysis in SOCOM++

The Ontology Rendition Step: after analysing the semantics of the given source and target ontology, the next step in the SOCOM++ prototype is ontology rendition. To achieve this, the *AOLT selection* process chooses the AOLT results in a specified mapping environment according to the *configurations* (see figure 5-3). Once AOLT results are determined, the Jena framework is used to render the converted source ontology (i.e. containing resources with translated labels and in original structure). A UML class diagram illustrating the rendition process is presented in figure 5-7.

As shown in figure 5-7, upon initiation, the *ontologyRendition* class initiates the *Socom* class which loads the property configuration and checks its validity. The validation ensures that the `properties.xml` file contains meaningful configurations for the system. For example, only one of the `<entry key="correctnessOptimise"/>` element and the `<entry key="completenessOptimise"/>` element can be set to `true` at a time (this is explained previously), or the value for the cut-off point when using the pseudo feedback in the `<entry key="threshold"/>` element must be between zero and one (this is explained in detail in section 5.4.3.3). Code snippet shown in appendix C, section C.4, figure C-15 illustrates how the validation is conducted. With a successful property validation, the *ExecutionFactory* class is initiated next that contains a collection of run methods implemented specifically to property configurations (they are discussed in detail in the remaining sections of this chapter).

For an entity in O_1, its candidate translations and their synonyms are compared to what is stored in the O_2 analysis. The *SemanticComparison* class is called to compare a given character string (i.e. a label) to the character strings (i.e. a set of labels) stored in the O_2 analysis, using string comparison technique (discussed in chapter 4, section 4.4). This process creates a record of candidate AOLT results via the *CandidateAOLTRecord* class. Figure 5-8 shows an example of the data that is contained in an AOLT record. The DTD used by the AOLT record can be found in appendix D, section D.3, figure D-6. As shown in figure 5-8, each `<Record/>` element contains a set of attributes that store information including the original source resource's label (value stored in the attribute `sourceValue`), its ID (value stored in the attribute `sourceID`), the candidate AOLT (value stored in the attribute `aoltValue`) and its ID (value stored in the attribute `aoltID`), how this candidate AOLT was concluded (value stored in the attribute `type`) and where the translation came from (value stored in the attribute `media`).

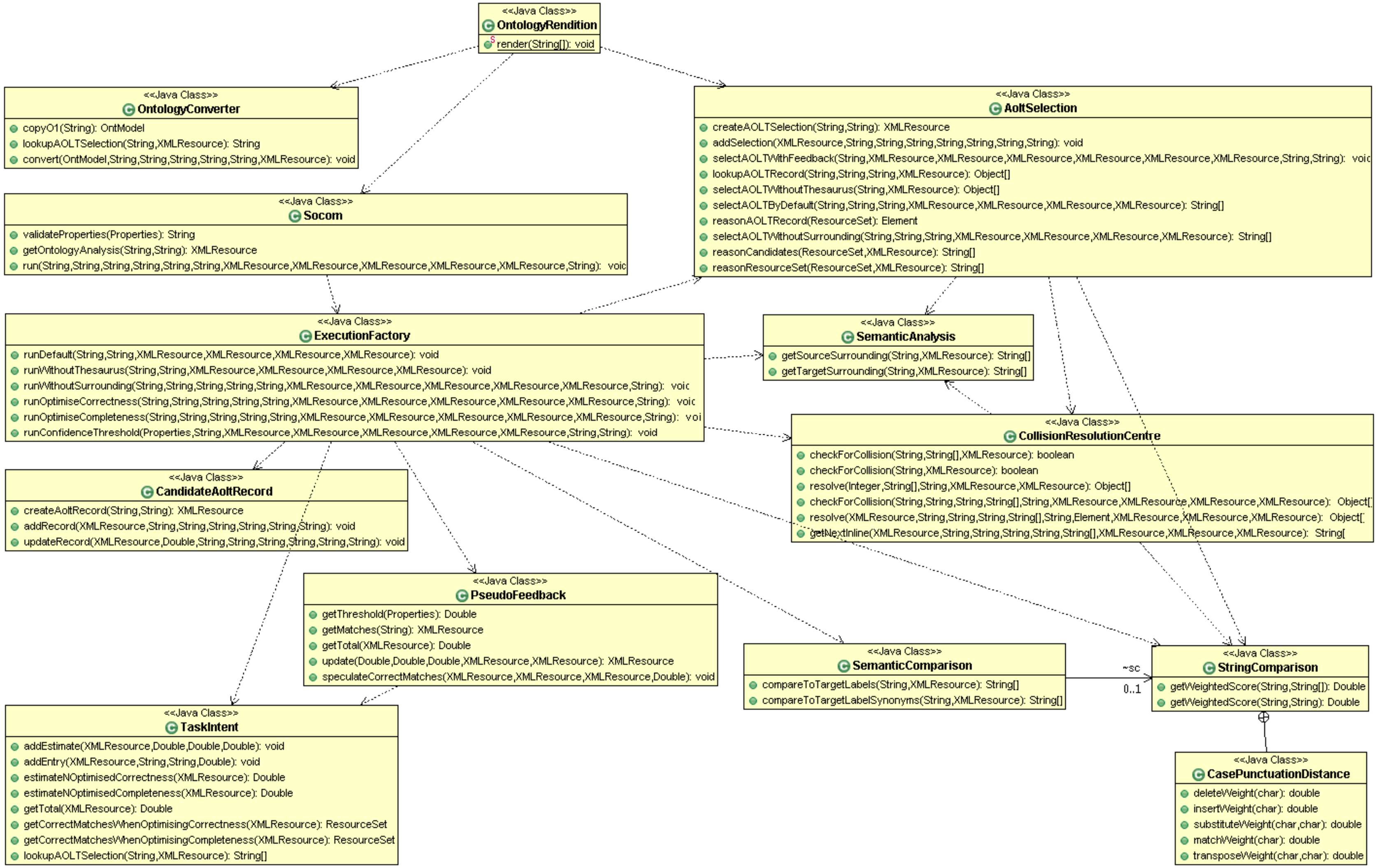


Figure 5-7. UML Class Diagram of Ontology Rendition in SOCOM++

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AOLTRecord SYSTEM "AOLTRecord.dtd">
<AOLTRecord>
    …
    <Record aoltID="CLS15" aoltValue="Organization" sourceID="CLS0" sourceValue="院所" media="BHT" type="2"/>
    <Record aoltID="SYN2-CLS22" aoltValue="establishment" sourceID="CLS0" sourceValue="院所" media="BHT" type="4"/>
    <Record aoltID="CDD0-CLS0" aoltValue="Institutions" sourceID="CLS0" sourceValue="院所" media="google" type="6"/>
    <Record aoltID="CDD1-CLS0" aoltValue="Institute" sourceID="CLS0" sourceValue="院所" media="bing" type="6"/>
    …
    <Record aoltID="DTP66" aoltValue="email" sourceID="DTP103" sourceValue="电子邮件" media="both" type="1"/>
    <Record aoltID="SYN1-DTP66" aoltValue="e-mail" sourceID="DTP103" sourceValue="电子邮件" media="both" type="3"/>
    <Record aoltID="SYN0-DTP66" aoltValue="electronic_mail" sourceID="DTP103" sourceValue="电子邮件" media="BHT" type="4"/>
    <Record aoltID="SYN2-DTP66" aoltValue="electronic_communication" sourceID="DTP103" sourceValue="电子邮件" media="BHT" type="4"/>
    <Record aoltID="CDD0-DTP103" aoltValue="E-mail" sourceID="DTP103" sourceValue="电子邮件" media="both" type="5"/>
    …
</AOLTRecord>
```

Figure 5-8. An Example of AOLT Record

There are six approaches to generate a candidate AOLT as summarised in table 5-1, discussed next.

- Type 1 denotes a match[93] found between a candidate translation (from O_1 analysis) and a target label (from O_2 analysis), whereby the target label is stored in the attribute `aoltValue` and its ID from the O_2 analysis is stored in the attribute `aoltID`.
- Type 2 illustrates a match between a synonym of a candidate translation and a target label. The target label is stored in the attribute `aoltValue` and its ID from the O_2 analysis is stored in the attribute `aoltID`.
- Type 3 refers to matches found between a candidate translation and a target label's synonym. This synonym is stored in the attribute `aoltValue`, and its ID from the O_2 analysis is stored in the attribute `aoltID`.
- Type 4 represents instances when matches are found between a synonym of a candidate translation and a synonym of a target label. The synonym of the target label is stored in the attribute `aoltValue` and its ID from the O_2 analysis is stored in the attribute `aoltID`.
- When the incorporated MT tools agree on the translation for a source label, this is stored as type 5 candidate AOLT. The agreed candidate translation

[93] A *match* in the context of storing candidate AOLT results refers to a pair of labels that has zero edit distance when the white spaces and character cases in them are ignored.

is stored in the attribute aoltValue and one of the IDs (from one of the MT tools) assigned during the O_1 analysis is stored in the aoltID.

- Type 6 refers to machine-generated candidate translations that differ (i.e. the case/space-insensitive edit distance between them is greater than zero) from one another. Each candidate translation is stored in a <Record/> element, with IDs that were assigned during the O_1 analysis. Note that type 6 conclusions can only exist with the absence of a type 5 conclusion in the AOLT record. When a type 5 conclusion is recorded, it implies there are two type 6 conclusions in the AOLT record which could also be stored. However, this is considered as redundant data in SOCOM++ as they do not add additional candidate AOLT results to the record.

Table 5-1. Types of Candidate AOLT Results

<table>
<tr><th>Type</th><th>O₁ Analysis</th><th></th><th>O₂ Analysis</th><th>Candidate AOLT</th></tr>
<tr><td>1</td><td>candidate translation</td><td rowspan="4">matches</td><td>target label</td><td>target label</td></tr>
<tr><td>2</td><td>candidate translation’s synonym</td><td>target label</td><td>target label</td></tr>
<tr><td>3</td><td>candidate translation</td><td>target label’s synonym</td><td>target label’s synonym</td></tr>
<tr><td>4</td><td>candidate translation’s synonym</td><td>target label’s synonym</td><td>target label’s synonym</td></tr>
<tr><td>5</td><td>MT agreed candidate translation</td><td>-</td><td>-</td><td>MT agreed candidate translation</td></tr>
<tr><td>6</td><td>MT disagreed candidate translation</td><td>-</td><td>-</td><td>each candidate translation</td></tr>
</table>

Type 1 to 6 candidate AOLT results are ordered in terms of the strongest to the weakest type of match in table 5-1. In the example shown in figure 5-8, the source label 院所 with ID CLS0 has four candidate AOLT results. The first candidate: Organization is derived from the BHT (the Big Huge Thesaurus API) via the type 2 match, has the CLS15 ID which was assigned during the O_2 analysis. A second candidate: establishment is also derived from the BHT via type 4 conclusion, has the SYN2-CLS22 ID in the O_2 analysis. A third candidate: Institutions is derived from google (the GoogleTranslate API) with ID CDD0-CLS0 (assigned during O_1 analysis) which differs from a fourth candidate: Institute which was returned from bing (the Microsoft Translator API) with ID CDD1-CLS0 (assigned during O_1 analysis).

After the AOLT record is prepared, the *ExecutionFactory* class initiates the *AoltSelection* class which begins the selection of the final AOLT results, as shown in figure 5-7. The *AoltSelection* class is responsible for choosing the AOLT results, solving any translation collisions that may occur via the *CollisionResolutionCentre* class and storing the final AOLT results in the database. (Note that how collisions are solved depends on the SOCOM++ configuration, six trials with six different resolution strategies are discussed later in section 5.4.) To access the candidate AOLT results, the *AoltSelection* class issues `XQuery` and `XPathQuery` via the XML:DB 1.0 API to the AOLT record in the database. The code snippet in appendix C, section C.4, figure C-16 presents an example of using `XQuery` via the XML:DB API. An example of querying the AOLT record using `XPathQuery` via the XML:DB API is presented in appendix C, section C.4, figure C-17. To solve collisions, the *CollisionResolutionCentre* class needs to determine which entity should keep the collided term and which alternative translation should be given to the other entity, by comparing semantic surroundings via the *SemanticAnalysis* class. The comparisons between character strings (i.e. a label vs. another label) and groups of character strings (i.e. a set of labels vs. another set of labels) are achieved by the *StringComparison* class, which is implemented in the same way as prototype one (see chapter 4, section 4.4) via the LingPipe API. Recall there are six approaches to conclude a candidate AOLT result - they are prioritised during the AOLT selection. For example, in the absence of a type 1 conclusion, use the type 2 candidate AOLT; if it causes collision or simply does not exist, use the type 3 candidate AOLT and so on. The AOLT selection algorithm varies depending on what resources are available to the system. This is discussed in detail through six trial experiments presented in the remaining sections of this chapter. The final AOLT results are stored in the eXsit DB, figure 5-9 presents an example of the AOLT selection. Each `<AOLT/>` element contains the attribute `sourceID` (the ID of the source label assigned during O_1 analysis), the attribute `media` (the translation source used to pin down the final AOLT for the source label), the attribute `type` (the AOLT conclusion type as discussed previously), the attribute `source` (the original URI for the resource with the given source label) and the attribute `translation` (the URI of the resource in the converted source ontology which contains a new base URI and translated label identifiers). The DTD used for the AOLT selections can be found in appendix D, section D.3, figure D-7.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AOLTSelection SYSTEM "AOLTSelection.dtd">
<AOLTSelection>
    …
    <AOLT sourceID="CLS-9" media="both" type="5" source="http://kdeg.cs.tcd.ie/CSWRC#经理
    " translation="http://kdeg.cs.tcd.ie/CSWRC/translated#Manager"/>
    <AOLT sourceID="CLS-12" media="both" type="1" source="http://kdeg.cs.tcd.ie/CSWRC#副
    教授" translation="http://kdeg.cs.tcd.ie/CSWRC/translated#Associate_Professor"/>
    <AOLT sourceID="CLS-30" media="google" type="3" source="http://kdeg.cs.tcd.ie/CSWRC#
    论文" translation="http://kdeg.cs.tcd.ie/CSWRC/translated#paper"/>
    <AOLT sourceID="OBP-57" media="BHT" type="2" source="http://kdeg.cs.tcd.ie/CSWRC#工作
    者" translation="http://kdeg.cs.tcd.ie/CSWRC/translated#Person"/>
    <AOLT sourceID="OBP-74" media="BHT" type="4" source="http://kdeg.cs.tcd.ie/CSWRC#成员
    " translation="http://kdeg.cs.tcd.ie/CSWRC/translated#associate"/>
    <AOLT sourceID="DTP-115" media="bing" type="6" source="http://kdeg.cs.tcd.ie/CSWRC#
    国际标准刊号"
    translation="http://kdeg.cs.tcd.ie/CSWRC/translated#International_standards_call_num
```

```
    ber"/>
    …
</AOLTSelection>
```

Figure 5-9. An Example of Final AOLT Results

As shown in figure 5-7, once AOLT results are selected for all the source labels, the *OntologyConverter* class is called to generate O_1' by looking up the AOLT selection stored in the database. The Jena Framework is implemented to construct the converted ontology in the target natural language. This process is previously demonstrated by the code snippet shown in appendix C, section C.2, figure C-1 and figure C-2.

The Ontology Mapping Step: upon the creation of the O_1' ontology, MOM techniques are applied to generate matches between O_1' and O_2 by using the Alignment API (this is the same with SOCOM). As figure 5-10 illustrates, the *Map* class is initiated that calls the *MatchingAlgorithmFactory* class which contains a collection of eight matching algorithms provided by the Alignment API.

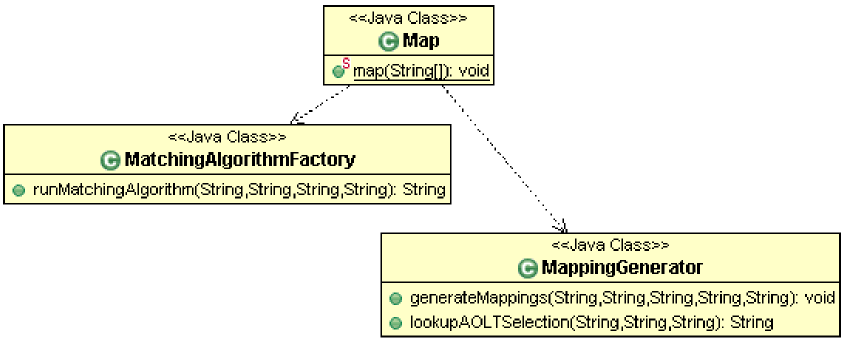


Figure 5-10. Class Diagram of Mapping Generation

The matches generated from this class however, consist of pairs of matched entities from O_1' and O_2 in the alignment format[94]. For example:

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'
         xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
         xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
         xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'>
  <Alignment>
    …
    <map>
      <Cell>
          <entity1
          rdf:resource='http://kdeg.cs.tcd.ie/CSWRC/translated#Assistant_Professor'/>
          <entity2
          rdf:resource='http://annotation.semanticweb.org/2004/iswc#Associate_Professor'
          />
          <relation>=</relation>
          <measure
          rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8157657657657658</meas
          ure>
      </Cell>
    </map>
    …
  </Alignment>
</rdf:RDF>
```

[94] http://oaei.ontologymatching.org/2009/align.html

To generate the final CLOM results between O_1 and O_2, the *MappingGenerator* class looks up the AOLT selection (e.g. figure 5-9) from the database and replaces all the O_1' entities (in target natural language) with O_1 entities (in source natural language). This is the same approach as was taken with the baseline system and SOCOM. The previous matches are converted now as:

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
<rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'
         xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
         xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
         xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/alignment#'>
  <Alignment>
    …
    <map>
      <Cell>
          <entity1 rdf:resource= http://kdeg.cs.tcd.ie/CSWRC#副教授'/>
          <entity2
          rdf:resource='http://annotation.semanticweb.org/2004/iswc#Associate_Professor'
          />
          <relation>=</relation>
          <measure
          rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>0.8157657657657658</meas
          ure>
      </Cell>
    </map>
    …
</Alignment>
</rdf:RDF>
```

5.4. SOCOM++ Evaluation

This section presents the evaluation of SOCOM++, which aims to demonstrate the impact of different configurations on the final mappings generated. A total of six trials have been carried out. The flexibility of SOCOM++ is demonstrated through these trials, with emphasis placed on adjusting the inputs of the AOLT process in an effort to influence the matching outcome. The goal of these trials is to investigate the impact of each input on the AOLT outcome and how the CLOM results are consequently influenced. Also, scalability tests are carried out to investigate the execution time required to complete a simpler and a more complicated trial run. The six trials are not an exhaustive list of how SOCOM++ can be configured, but rather examples of typical adjustment on the AOLT selection process. The first three trials (discussed in section 5.4.2) focus on adjusting the inputs that are related to the given ontologies involved in a mapping scenario. The other three trials (discussed in section 5.4.3) focus on executing a second iteration of the AOLT process. Each trial focuses on one of the six inputs (discussed in section 5.3) of the AOLT process. An overview of the six trial experiments is presented in section 5.4.1. Section 5.4.2 presents the first three trials that focus on semantic adjustments. Section 5.4.3 presents another three trials that focus on the second iteration of the AOLT process. Section 5.4.4 discusses the conclusions drawn from the six trials undertaken. Finally, section 5.4.5 presents the scalability tests.

5.4.1. Overview of Trials

An overview of the configurations used in the six trials can be found in table 5-2. For a given CLOM scenario, inputs to the AOLT process are adjusted, with the exception of two: candidate translations (of O_1 labels) and O_2 labels. These two inputs are essential in any trial, because candidate translations must be available to bridge between the natural languages presented in O_1 and O_2; and O_2 labels must be consulted to realise the AOLT concept (since the AOLT process at core concerns selecting translations that are the same/similar with the labels in O_2).

Each trial has a different configuration of the inputs to the AOLT process. The evaluation of each configuration consists of two CLOM experiments, which were first used in the evaluation of SOCOM (discussed in chapter 4, section 4.5), namely, mapping the CSWRC ontology (in Chinese) to the ISWC ontology (in English) of the research domain and mapping the 101 ontology (in English) to the 206 ontology (in French) of the bibliography domain. These two CLOM experiments are used in the evaluation of each trial in SOCOM++. Such experimental setups (i.e. the same ontology pairs are used again) will continuously examine the AOLT process, since how AOLT results are achieved is the only difference between SOCOM++ and SOCOM.

Table 5-2. A Summary of SOCOM++ Trial Configurations

Configuration	Inputs of the AOLT Selection Process							
	Candidate Translations	Synonyms of Candidate Translations	O_1 Semantic Surroundings	O_2 Labels	Synonyms of O_2 Labels	O_2 Semantic Surroundings	2^{nd} Iteration	Cut-off Point
Trial 1	✓	✓	✓	✓	✓	✓	✗	n/a
Trial 2	✓	✗	✓	✓	✗	✓	✗	n/a
Trial 3	✓	✓	✗	✓	✓	✗	✗	n/a
Trial 4	✓	✓	✓	✓	✓	✓	✓	Conf.=1
Trial 5	✓	✓	✓	✓	✓	✓	✓	Conf.>0
Trial 6	✓	✓	✓	✓	✓	✓	✓	Conf.≥0.5

Trial one, two and three concern the adjustment on inputs that are related to the ontologies involved in a CLOM scenario. A second iteration of the AOLT process is not conducted in these three trials. Trial one (discussed in section 5.4.2.1) focuses on adjusting the execution constraint, whereby the default configuration is applied to achieve AOLT results. This default execution is essentially an enhanced SOCOM, where (as explained previously in section 5.2) the enhancement is that synonyms of candidate translations (of O_1 labels, in addition to O_2 labels and their synonyms) are included in the AOLT selection process. This trial thus investigates whether an increased candidate AOLT pool (through the added synonyms of the candidate translations) could improve the matching quality. Matches generated from trial one and their evaluations can be found at `root/SOCOM++Experiments/TrialOne/` on the DVD.

Trial two (discussed in section 5.4.2.2) focuses on adjusting the resource constraint property, whereby background resources such as thesauri are made unavailable to the AOLT selection process. This configuration thus does not include synonyms (of either O_2 labels or candidate translations of O_1 labels) in the AOLT selection process. This trial investigates the impact on the matching quality when the system only has access to the minimum amount of information that is naturally available (i.e. the semantics in the given ontologies - the labels in them and their semantic surroundings). Matches generated from trial two and their evaluations can be found at `root/SOCOM++Experiments/TrialTwo/` on the DVD.

Trial three (discussed in section 5.4.2.3) focuses on adjusting the embedded semantics that are available to the system, whereby semantic surroundings are not included in the AOLT process. This configuration draws AOLT conclusions from background knowledge alone (i.e. synonyms of O_2 labels and synonyms of candidate translations of O_1 labels) and investigates how the absence of semantic surroundings may impact on the matching outcome. Matches generated in trial three and their evaluations can be found at `root/SOCOM++Experiments/TrialThree/` on the DVD.

Trial four, five and six focus on carrying out a second iteration of the AOLT process using three different selection rationales to achieve AOLT results during the second iteration. The selection rationales are achieved through the optimising correctness task intent, the optimising completeness task intent and the pseudo feedback feature. As discussed in section 5.3, only one task intent or pseudo feedback can be in effect at a time. Trial four (discussed in section 5.4.3.1) focuses on adjusting the optimising correctness intent in the configuration. This trial investigates how a strict cut-off point in the initial iteration (i.e. only matches with 1.0 confidence levels are assumed to be correct) may impact on the matching outcome generated from the second iteration of the AOLT process. Matches generated from trial four and their evaluations can be found at `root/SOCOM++Experiments/TrialFour/` on the DVD.

Trial five (discussed in section 5.4.3.2) applies the optimising completeness task intent in the configuration, and investigates how the matching quality is effected in the second iteration when no cut-off point is applied in the assumption (i.e. all matches generated from the first iteration are assumed to be correct). This configuration effectively prioritises most frequently used selection rationales (i.e. MT media used) in the second iteration of the AOLT process according to their popularity in the first iteration. Matches generated from trial five and their evaluations can be found at `root/SOCOM++Experiments/TrialFive/` on the DVD.

Trial six (discussed in section 5.4.3.3) focuses on the pseudo feedback feature, which offers adjustment of the cut-off point on confidence levels. This feature allows the user to specify a threshold anywhere between the cut-off points used in trial four and trial five. (Note that trial four and five do not offer adjustable cut-off points.) In trial six, a threshold of 0.5 is applied (i.e. matches with at least 0.5 confidence levels from iteration one are assumed to be correct). This cut-off point was chosen as it is a natural division between 0.0 and 1.0, where equal to/greater than 0.5 indicates an incline towards confident, and less than 0.5 indicates an incline towards not confident. Matches generated from trial six and their evaluations can be found at `root/SOCOM++Experiments/TrialSix/` on the DVD.

Figure 5-11 presents an overview of the experiments carried out in the evaluation of SOCOM++. Experiment one (figure 5-11-a) requires the mapping of the Chinese CSWRC ontology to the English ISWC ontology. Experiment two (figure 5-11-b) concerns the mapping of the English 101 ontology to the French 206 ontology. In both experiments, eight MOM matching algorithms (provided by the Alignment API) have been applied to generate matches for the baseline system as well as SOCOM++. The generation of the gold standards for these experiments have been discussed previously (see chapter 4, section 4.5.1.1 and 4.5.2.1).

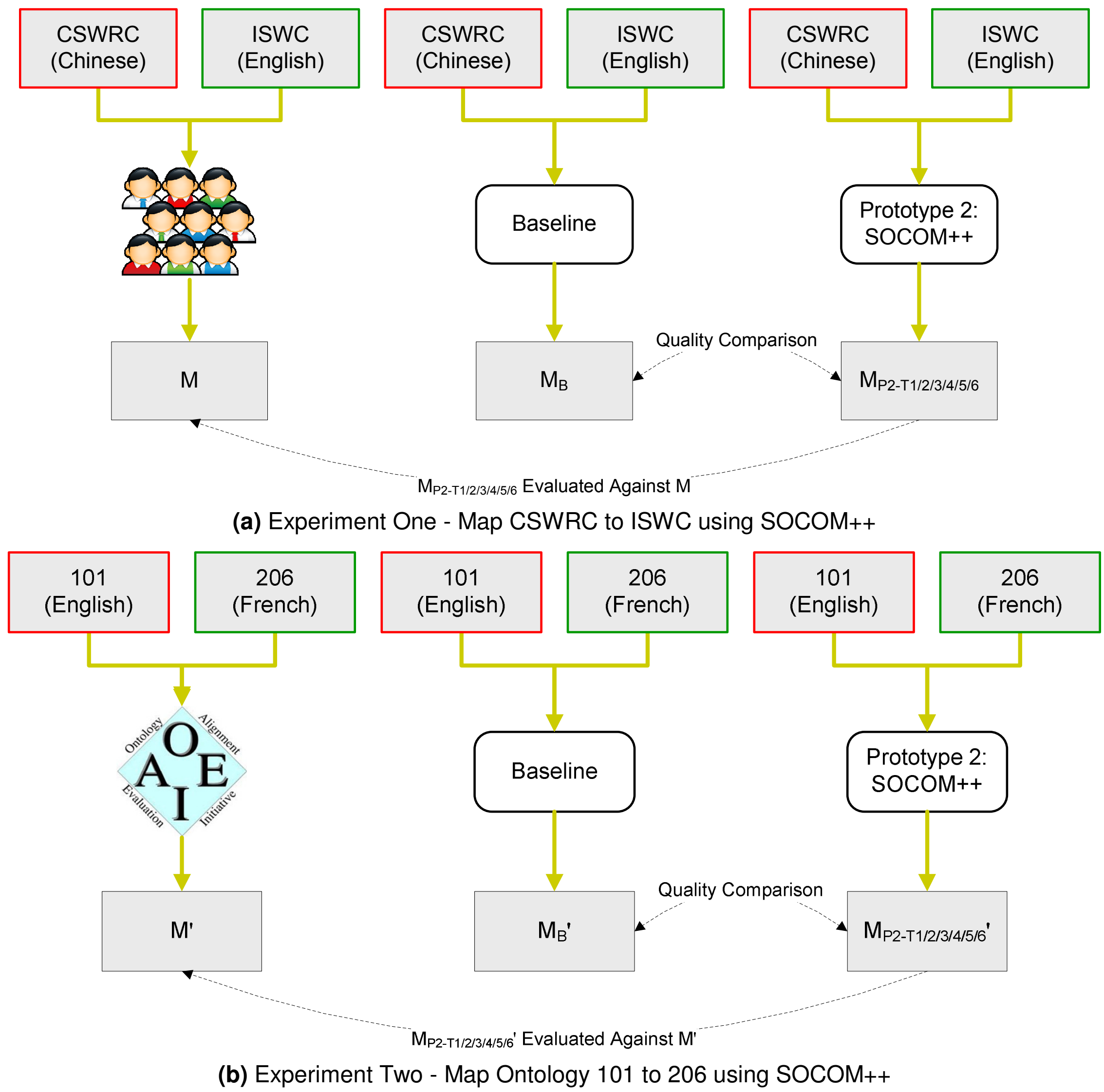


Figure 5-11. Experimental Setup of Trial One to Six

In experiment one (figure 5-11-a), M is the gold standard between the CSWRC ontology and the ISWC ontology. M_B is the matches generated by the baseline system, which contains eight sets of matches (each set is generated by a MOM algorithm). $M_{P2\text{-}T1/2/3/4/5/6}$ is the matches generated by the SOCOM++, where $M_{P2\text{-}T1}$ contains eight sets of matches generated from trial one, $M_{P2\text{-}T2}$ contains eight sets of matches generated from trial two and so on. $M_{P2\text{-}T1/2/3/4/5/6}$ is evaluated against the gold standard M, and compared to M_B. In experiment two, M' is gold standard between ontology 101 and 206. M_B' is the matches generated by the baseline system. $M_{P2\text{-}T1/2/3/4/5/6}$' refers to the matches generated by the SOCOM++, where $M_{P2\text{-}T1}$' contains eight sets of matches generated in trial one, $M_{P2\text{-}T2}$' contains eight sets of matches generated in trial two and so on. $M_{P2\text{-}T1/2/3/4/5/6}$' is evaluated against gold standard M', and compared to M_B'.

5.4.2. Three Trials to adjust Ontology Semantics

This section presents trial one (discussed in section 5.4.2.1), two (discussed in section 5.4.2.2) and three (discussed in section 5.4.2.3) that focus on the adjustment of ontology semantics during the AOLT process.

5.4.2.1. Trial One - adjust Execution Constraint

Trial one investigates whether the default AOLT process in SOCOM++ can improve the mapping quality compared to the baseline system, or even what was achieved by SOCOM (discussed in chapter 4, section 4.5). The setup of this trial is discussed in section 5.4.2.1.1, and the findings are presented in section 5.4.2.1.2.

5.4.2.1.1. Trial Setup

As discussed previously (in section 5.3), the system property: execution constraint can be set to either `true` or `false`. When it is set to `true`, all other property settings are ignored and the system is executed with its default configuration. The default configuration of the system makes use of all the resources that are available to aid the AOLT process, including the candidate translations, their synonyms, source semantic surroundings from the O_1 analysis and target labels, their synonyms, target semantic surroundings from the O_2 analysis. For each source label, its candidate translations and synonyms are compared to what is stored in the O_2 analysis and a record of candidate AOLT results are generated and stored as shown previously in figure 5-8.

When selecting the AOLT result for a source label, the system looks through the AOLT record for the lowest possible conclusion type (by issuing XPath queries - demonstrated by the code snippet shown in appendix C, section C.4, figure C-16). This is because lower conclusion types illustrate stronger matches to the data in O_2 analysis (see table 5-1 for conclusion types, how they are generated and what they represent). In the absence of a low conclusion type for a source label, its alternative candidate AOLT with a higher type would be selected. In the example shown in figure 5-8, the source label 院所 with ID `CLS15` does not have a type `1` candidate AOLT (i.e. when a candidate translation matches a target label), hence the type `2` candidate AOLT (i.e. when a candidate translation’s synonym matches a target label) would be selected as the AOLT. Note that more than one candidate AOLT with the same conclusion type may exist for a source label. An example can be seen in figure 5-8, the source label 院所 with ID CLS15 has two type 6 (i.e. when each MT tool gives a differing translation, and these translations do not match anything in the O_2 analysis) candidate AOLT results, and the source label 电子邮件 with ID DTP103 has two type 4 (i.e. when candidate translation's synonym matches target label's synonym) candidate AOLT results. When more than one candidate AOLT result with the same desired type are available to the selection process, in the case of type 1, 2, 3 and 4 candidates, the candidate AOLT that is most similar to the target surrounding is chosen as the most suitable AOLT. This is because these conclusion types are derived with association to the O_2 analysis. In the case of type 5 candidate AOLT results, no further comparison to semantic surrounding is necessary. This is because type 5 candidate AOLT results are a result of MT tools being in agreement for the translation of a source label, there can only be one type 5 record at most. In the case of having more than one type 6 candidate AOLT results being available, the selection process chooses the candidate AOLT that is most similar (using string comparison technique, discussed in chapter 4, section 4.4) to the source semantic surrounding in the O_1 analysis.

Collisions of AOLT results can occur when the aforementioned selection process chooses the same translation term (i.e. two character strings that are identical to one another) for two or more source labels, which must be resolved before storing of the final AOLT results in the database. The resolution strategies are summarised in table 5-3, which include 11 types of collisions as scenarios i to xi. To solve a collision between a pair of entities E_1 and E_2, their candidate types are checked. The entity whose candidate was concluded as having a lower type keeps the collided term as its AOLT result, and the other entity will then seek an alternative candidate from the AOLT record with the lowest possible type other than its current collided type, as demonstrated by the scenarios i to x in table 5-3.

For example, the best available candidate in the AOLT record for 院所 with ID CLS0 is Organization:

```
<Record aoltID="CLS15" aoltValue="Organization" sourceID="CLS0" sourceValue="
院所" media="BHT" type="2"/>
```

However, the best available AOLT candidate for another entity label 组织 with ID CLS3 is also Organization:

```
<Record aoltID="CLS15" aoltValue="Organization" sourceID="CLS3" sourceValue="
组织" media="both" type="1"/>
```

As the former is a type 2 record (i.e. concluded based on a match between a synonym of 院所's candidate translation and a target label - a weaker match in comparison) and the latter is a type 1 record (i.e. concluded based on a match between the candidate returned from MT and a target label - a stronger match in comparison), the second entity 组织 will keep `Organization` as its AOLT result and the first entity 院所 will seek the next best available translation from the AOLT record, e.g. a type 3 record; if type 3 candidate is not available or causes collision (either with the same entity or other entities), a type 4 record will be chosen and so on. If a pair of collided entities involve the same type of record, as demonstrated by scenario xi in table 5-3, the collided AOLT is compared to the semantic surrounding of E_1 and the semantic surrounding of E_2. The entity whose semantic surrounding is most similar (using string comparison technique) to the candidate AOLT will keep this collided term as its AOLT result, and the other entity will seek the next available AOLT in the same fashion as discussed above.

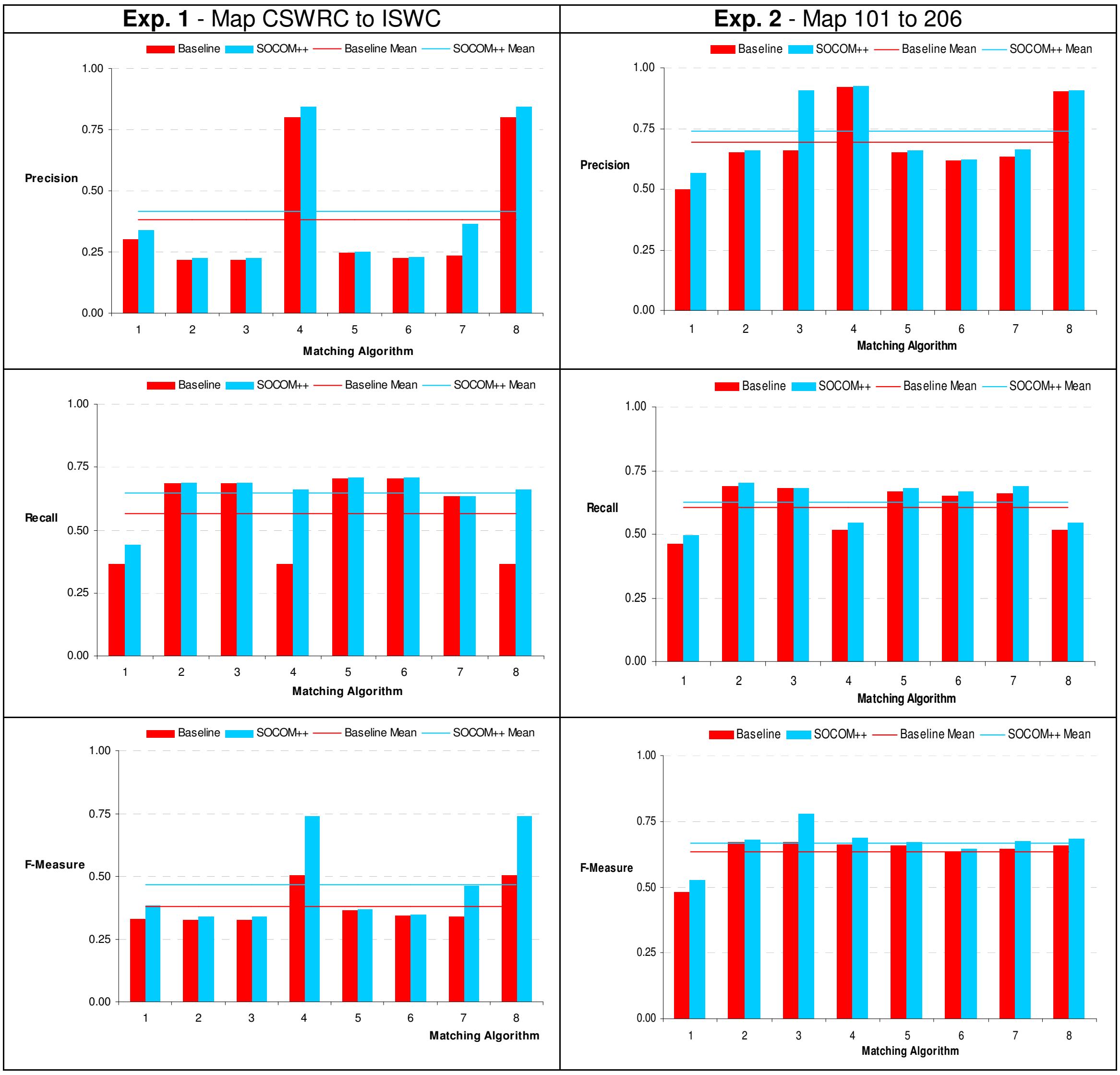
Table 5-3. Collision Resolution in SOCOM++ Trial One

Collision Scenario	Candidate AOLT		Solution
	E_1	E_2	
i	type = 1	type = 2, 3, 4, 5 or 6	E_1 keeps the collided AOLT; E_2 seeks alternative AOLT with lowest possible type other than the current type.
ii	type = 2	type = 3, 4, 5 or 6	
iii	type = 3	type = 4, 5 or 6	
iv	type = 4	type = 5 or 6	
v	type = 5	type = 6	
vi	type = 2, 3, 4, 5 or 6	type = 1	E_2 keeps the collided AOLT; E_1 seeks alternative AOLT with the lowest possible type other than the current type.
vii	type = 3, 4, 5 or 6	type = 2	
viii	type = 4, 5 or 6	type = 3	
ix	type = 5 or 6	type = 4	
x	type = 6	type = 5	
xi	E_1 type = E_2 type		Entity that is most similar to source surrounding keeps the collided AOLT; the other entity seeks alternative AOLT with the lowest possible type other than the current type.

If collisions remain unsolved after all available candidates in the AOLT record have been investigated for a source label, a unique integer is attached to the collided term as the AOLT for this entity (to break out from the recursive process which seeks the next best AOLT result). This is achieved in the same way as the baseline system (see chapter 3, section 3.3.2) and SOCOM (see chapter 4, section 4.4). SOCOM++ with the default AOLT process discussed in this section is evaluated in the following section.

5.4.2.1.2. Findings and Analysis

A summary of the findings on precision, recall and f-measure from trial one can be found in figure 5-12. The left column in figure 5-12 contains the findings from experiment one - mapping the CSWRC ontology to the ISWC ontology. The right column contains the findings from experiment two - mapping the 101 ontology to the 206 ontology. These findings are generated when a match is considered correct as long as it is included in the gold standard regardless of its confidence level.



Legend: 1 NameAndPropertyAlignment 5 SMOANameAlignment
2 StrucSubsDistAlignment 6 SubsDistNameAlignment
3 ClassStructAlignment 7 EditDistNameAlignment
4 NameEqAlignment 8 StringDistAlignment

Figure 5-12. Precision, Recall and F-Measure found in Trial One

In experiment one, improvements in precision can be seen across all eight matching algorithms when SOCOM++ is applied. This finding indicates that no matter which matching algorithm was applied, the default configuration of SOCOM++ was able to generate more number of correct matches in this experiment than the baseline system. The average precision in $M_{P2\text{-}T1}$ is 0.4155, which is an average improvement of 9.54% compared to the average precision of M_B (at 0.3793). A similar finding can be seen in the recall scores generated: when SOCOM++ is applied, equal (in the case of the *EditDistNameAlignment* algorithm) or higher (in the case for all other algorithms) recall is found in this experiment with respect to the baseline system. Particularly in the case of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm, substantial higher recall scores are obtained in this experiment. This is because both algorithms are lexicon-based and employ strict string comparison techniques when concluding entity matches[95]. With the selection of AOLT results for resource labels in SOCOM++, the completeness of the matches has been greatly improved for these algorithms. An average recall of 0.6488 is found in $M_{P2\text{-}T1}$, which is an average improvement of 15.04% compared to M_B (at 0.5640). The overall matching quality is illustrated by the f-measure achieved. Higher f-measure can be seen in all matching algorithms when the SOCOM++ is applied. This suggests the quality of the matches generated by SOCOM++ is higher than those generated by the baseline system. On average, an f-measure of 0.4654 is found in $M_{P2\text{-}T1}$, which is a 23.06% improvement over M_B (at 0.3782). The p-value derived from the paired t-test on the f-measure scores collected in $M_{P2\text{-}T1}$ and M_B is 0.044. At a significance level of α=0.05, this p-value rejects the null hypothesis (being that there is no difference between $M_{P2\text{-}T1}$ and M_B) and supports the finding that matches generated by SOCOM++ are of higher quality than those generated by the baseline system in the CLOM scenarios studied.

In experiment two, improvements in precision can be seen across all eight MOM algorithms. On average, the baseline system achieved 0.6918 precision in this experiment, and a higher precision of 0.7394 was achieved by SOCOM++. This is an average improvement of 6.88%. This result shows that a larger number of correct matches were generated by the SOCOM++ in this experiment. More visible improvements can be seen in the recall scores generated. A mean recall of 0.6057 was found in the baseline system, and a higher mean of 0.6261 was found in SOCOM++. This is an average improvement of 3.37%. This finding shows that the matches generated by SOCOM++ were more complete than those generated via the baseline system in this experiment. A similar trend can be seen in f-measure. Improvement in matching quality is visibly shown in all matching algorithms executed. On average, an f-measure of 0.6347 is found in M_B', whereas a higher f-measure of 0.6684 was found in $M_{P2\text{-}T1}'$. This is an average improvement by 5.31% in the overall quality of the matches generated. The p-value derived from paired t-test carried out on the f-measure

[95] Only matches with 1.0 confidence levels are generated by these algorithm since only entities with identical labels are matched (e.g. *assistant_professor* and *Assistant_Professor* is not a match because these character strings are not identical).

scores yield 0.023, suggesting that these differences in the results generated are statistically significant. This provides further evidence that the matches generated from SOCOM++ are of higher quality than those generated from the baseline system.

The confidence levels of the matches generated are investigated next in the evaluation. The confidence means and standard deviations are calculated in both experiments. These results are shown in table 5-4. Scatter plots generated using this data can be found in appendix E, section E.1, figure E-1.

Table 5-4. Confidence Data from Trial One

Exp.	Matching Technique	Baseline		SOCOM++ Trial 1 – Adjust Execution constraint	
		St. Dev.	Mean	St. Dev.	Mean
1	1 NameAndPropertyAlignment	0.1014	0.9374	0.0544	0.9872
	2 StrucSubsDistAlignment	0.2505	0.7505	0.2246	0.8186
	3 ClassStructAlignment	0.2505	0.7505	0.0160	0.9969
	5 SMOANameAlignment	0.0582	0.9649	0.0160	0.9969
	6 SubsDistNameAlignment	0.1618	0.9041	0.0453	0.9911
	7 EditDistNameAlignment	0.0123	0.9909	0.0112	0.9969
	Avg.	**0.1391**	**0.8830**	**0.0613**	**0.9646**
2	1 NameAndPropertyAlignment	0.0909	0.9674	0.0881	0.9774
	2 StrucSubsDistAlignment	0.1509	0.9059	0.1485	0.9233
	3 ClassStructAlignment	0.1545	0.9440	0.1140	0.9577
	5 SMOANameAlignment	0.1556	0.9431	0.0925	0.9664
	6 SubsDistNameAlignment	0.1541	0.9372	0.1791	0.9245
	7 EditDistNameAlignment	0.0179	0.9913	0.0165	0.9935
	Avg.	**0.1207**	**0.9481**	**0.1065**	**0.9571**

In experiment one, confidence levels have been improved (i.e. increased confidence mean and decreased standard deviation) by SOCOM++ for all algorithms. In experiment two, with the exception of the *SubsDistNameAlignment* algorithm, all other algorithms showed improved confidence levels when using SOCOM++. On average, the average confidence mean in $M_{P2\text{-}T1}$ is increased by 9.24% (to 0.9646), and the average standard deviation is decreased by 55.93% (to 0.0613) compared to M_B. The average confidence mean of M_B' (at 0.9481) is improved by 0.95% (to 0.9571) in $M_{P2\text{-}T1}'$. The average standard deviation of M_B' (at 0.1207) is decreased by 11.76% in $M_{P2\text{-}T1}'$ (to 0.1065). These results denote that the matches generated by SOCOM++ are not only more confident but their confidence levels are also less dispersed.

In summary, it is shown through the evaluation that SOCOM++ trial one (i.e. with default configuration) exceeds the baseline system in terms of precision, recall, f-measure as well as confidence level means and standard deviations. However, when compared to SOCOM, the improvement of SOCOM++ is not always evident. Table 5-5 presents the key findings from the baseline system (M_B and M_B'), SOCOM (M_{P1} and M_{P1}') and SOCOM++ trial one ($M_{P2\text{-}T1}$ and $M_{P2\text{-}T1}'$). In experiment one, improvement in

SOCOM++ trial one is evident in all measures except precision compared to SOCOM. In experiment two, improvement in SOCOM++ trial one is evident in all measures expect recall and confidence mean. This partial improvement can be understood as: with an increased candidate AOLT pool, it consequently comes with an increased risk of selecting incorrect AOLT results. In fact, this is later shown in trial two (discussed in section 5.4.2.2): when synonyms are removed (leading to a much smaller candidate AOLT pool), the mapping quality is not necessarily decreased as the AOLT process is more likely to select translations that are used by the target ontology (hence more exact matches). Nonetheless, it is worth noting that the default SOCOM++ configuration has further improved matching quality on several aspects compared to SOCOM. It is thus motivating for this author to explore other configurations, discussed next.

Table 5-5. Key Findings of Baseline, SOCOM and SOCOM++ Trial One

Evaluations		Baseline	SOCOM	SOCOM++ Trial 1 (default configuration)
Exp.1	Precision	0.3793	0.4367	0. 4155
	Recall	0.5640	0.5854	0.6488
	F-Measure	0.3782	0.4146	0.4654
	Confidence Level Mean	0.8830	0.8962	0.9646
	Confidence Level St.Dev.	0.1391	0.1239	0.0613
Exp.2	Precision	0.6918	0.7084	0.7394
	Recall	0.6057	0.6353	0.6261
	F-Measure	0.6347	0.6621	0.6684
	Confidence Level Mean	0.9481	0.9640	0.9571
	Confidence Level St.Dev.	0.1207	0.1110	0.1065

5.4.2.2. Trial Two - adjust Resource Constraint

Trial two investigates the effects of restricted background semantics (e.g. when thesauri are unavailable to the AOLT process) on the matching quality. The experimental setup is discussed in section 5.4.2.2.1, and the findings are presented in section 5.4.2.2.2.

5.4.2.2.1. Trial Setup

In specialised domains (e.g. medicine), it may be the case that there simply is few other ways to express certain concepts, or it may be the case that background resources which synonyms can be extracted from are simply not accessible. This trial aims to investigate how the matching outcome is affected given a lack of background semantics.

As discussed previously (see section 5.3), resource constraint is modelled by two `entry` elements (see figure 5-2), one with `key` attribute `translationSynonym` and the other with `key` attribute `targetSynonym`. Both elements can be configured to either `true` or `false`. In trial two, both elements are configured to `false`, which illustrates a case where thesauri are unavailable[96]. This means that the synonyms of candidate translations for source labels and the synonyms for target labels are not included to the AOLT process. As a result, there will only be type 1, 5 and 6 candidate AOLT results (see table 5-1 for types of candidate AOLT results), but no type 2, 3 or 4 candidates in the AOLT record. When selecting AOLT results, the system looks up the AOLT record and prioritises candidates with lower `type` attributes. If a type 1 candidate is available for a source label, it is selected as the AOLT result; in the absence of type 1 candidate, a type 5 candidate would be selected as the final AOLT and so on.

A summary of the strategies used to resolve collisions in trial two is presented in table 5-6. For a pair of collided entity E_1 and E_2, their AOLT results' respective candidate types are checked first. The entity with the lower type keeps the collided term as its final AOLT result, and the other entity seeks an alternative translation, as demonstrated by the scenarios i, ii, iii and iv in table 5-6. If both entities end with the same term based an equal type (as demonstrated by scenario v in table 5-6), the entity with semantic surrounding that is most similar (i.e. lowest aggregated edit distance) to that of the source label will keep the collided term as its AOLT result, and the other entity must seek an alternative translation (i.e. another translation with the lowest possible type other than the current type).

Table 5-6. Collision Resolution in Trial Two

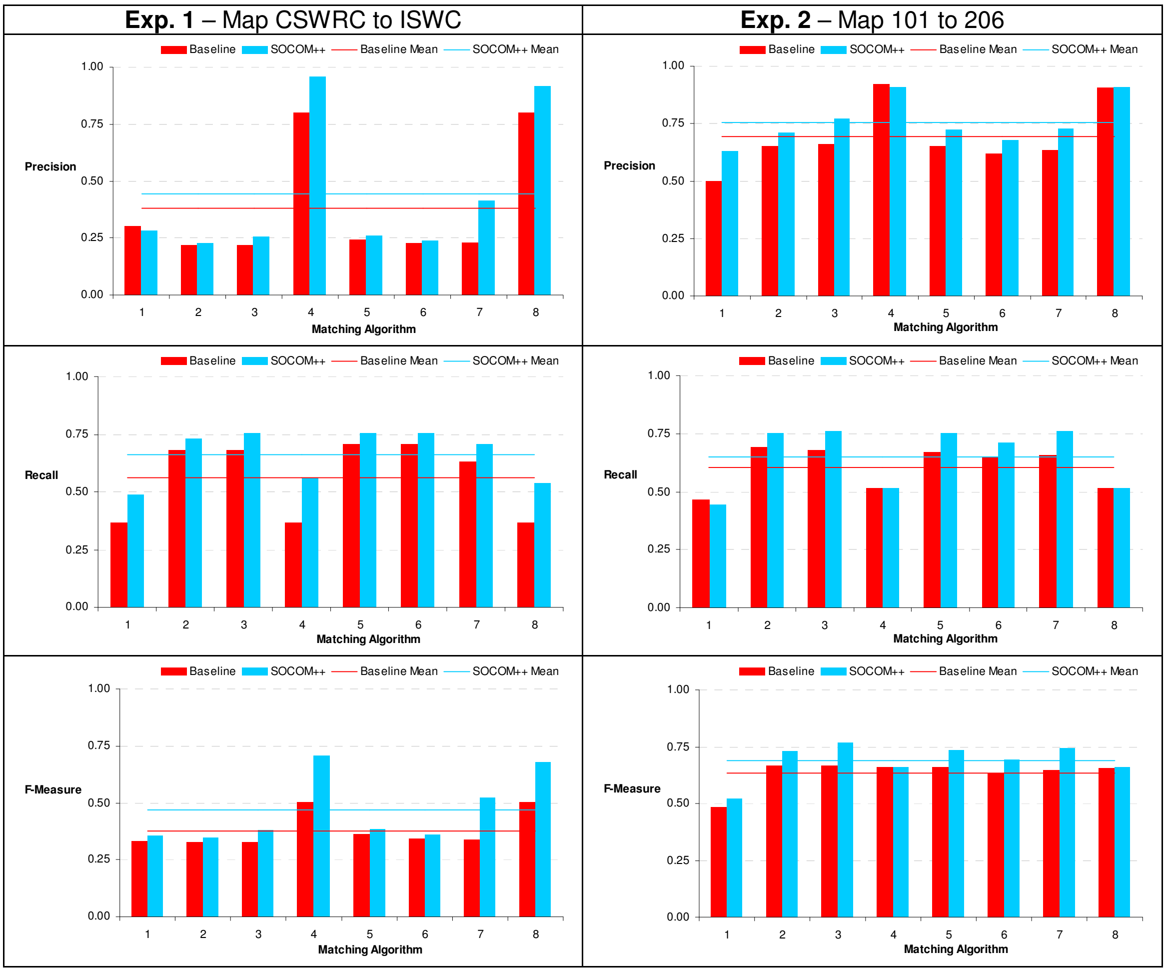
<table>
<tr><th rowspan="2">Collision Scenario</th><th colspan="2">Candidate AOLT</th><th rowspan="2">Solution</th></tr>
<tr><th>E₁</th><th>E₂</th></tr>
<tr><td>i</td><td>type = 1</td><td>type = 5 or 6</td><td rowspan="2">E1 keeps the collided AOLT; E2 seeks alternative AOLT with lowest possible type other than the current type.</td></tr>
<tr><td>ii</td><td>type = 5</td><td>type = 6</td></tr>
<tr><td>iii</td><td>type = 5 or 6</td><td>type = 1</td><td rowspan="2">E2 keeps the collided AOLT; E1 seeks alternative AOLT with the lowest possible type other than the current type.</td></tr>
<tr><td>iv</td><td>type = 6</td><td>type = 5</td></tr>
<tr><td>v</td><td colspan="2">E1 type = E2 type</td><td>Entity that is most similar to source surrounding keeps the collided AOLT; the other entity seeks alternative AOLT with the lowest possible type other than the current type.</td></tr>
</table>

If all alternatives have been explored and none are suitable (i.e. cause further collisions, or simply do not exist in the requested AOLT type), a unique integer is attached to the collided term for the entity with no more appropriate alternatives. This strategy is used in the baseline system, SOCOM and SOCOM++ trial one. Trial two is evaluated in experiments discussed in section 5.4.1, and the findings are presented next.

5.4.2.2.2. Findings and Analysis

[96] Note that the execution constraint, i.e. `<entry key="default"/>` must be set to `false`, in order for the system to account settings of other properties (see figure 5-2).

The precision, recall and f-measure scores generated in experiment one (mapping the CSWRC ontology to the ISWC ontology, shown in the left column) and experiment two (mapping ontology 101 to ontology 206, shown in the right column) can be seen in figure 5-13. These scores are generated when a match is considered correct so long it is included in the gold standard regardless of its confidence level.



Matching Algorithms: 1 NameAndPropertyAlignment 5 SMOANameAlignment
2 StrucSubsDistAlignment 6 SubsDistNameAlignment
3 ClassStructAlignment 7 EditDistNameAlignment
4 NameEqAlignment 8 StringDistAlignment

Figure 5-13. Precision, Recall, F-Measure in SOCOM++ Trial Two

In experiment one, with the exception of the *NameAndPropertyAlignment* algorithm, all other matching algorithms experienced some degree of improvement on precision. On average, a precision of 0.3793 was achieved by M_B, and a higher precision of 0.4437 was achieved by $M_{P2\text{-}T2}$. This is an average improvement of 16.98% on the number of correct matches generated using the SOCOM++ trial two configuration. Significant improvements can be seen in the recall scores generated by all eight matching algorithms. An average recall of 0.5640 was found in M_B where as an average of 0.6616 was found in $M_{P2\text{-}T2}$. This is a 17.30% improvement on the completeness of the correct matches when using SOCOM++ with the trial two configuration. Overall, improvement can be seen in all matching algorithms through the f-measure scores generated. An average f-measure of 0.3782 was found in M_B, and an average of 0.4674 was found in $M_{P2\text{-}T2}$. This is an improvement of 23.59%. This finding is further supported by the p-value found in the paired t-test of the f-measure scores generated by the two systems. At a p-value of 0.019, the paired t-test rejects null hypothesis of there being no difference between the two systems.

In experiment two, with the exception of the *NameEqAlignment* algorithm, all other algorithms generated higher precision scores in $M_{P2\text{-}T2}$'. An average precision of 0.7569 was found in $M_{P2\text{-}T2}$', which is an improvement by 9.41% compared to the baseline system (with an average precision of 0.6918). The average recall score is also improved when the SOCOM++ trial two configuration was applied, which yielded an average recall of 0.6521 - an improvement by 7.66% compared to the baseline system (with an average recall of 0.6057). Except for the *NameAndPropertyAlignment* algorithm and the *StringDistAlignment* algorithm, all other algorithms generated equal or higher recall scores in this trial as shown in figure 5-13-b. The f-measure scores reveal that with the exception of the *NameEqAlignment* algorithm, all other algorithms were able to improve the overall matching quality in $M_{P2\text{-}T2}$'. An average f-measure of 0.6886 was found in SOCOM++ trial two, which is an improvement of 8.49% compared to the baseline system (with an average f-measure of 0.6347). The p-value generated from the paired t-test on f-measure score is 0.006, which supports the statistical significance of the findings so far.

The evaluation carried out on the confidence levels can be found in table 5-7. Scatter plots generated using this data can be found in appendix E, section E.2, figure E-2. In experiment one, the confidence means are increased and the standard deviations are decreased for all matching algorithms in $M_{P2\text{-}T2}$. On average, a mean of 0.9326 was found in SOCOM++ trial two, which is an improvement by 5.62% compared to the baseline system (at a mean of 0.8830). An average standard deviation of 0.1088 was found in $M_{P2\text{-}T2}$, which is a decrease by 21.78% compared to M_B (with a standard deviation of 0.1391). In experiment two, the average mean and standard deviation have not been improved in this trial. Data in table 5-7 shows that matches in M_B ' were more confident and with less dispersed confidence levels than matches in $M_{P2\text{-}T2}$'.

Table 5-7. Confidence Data from Trial Two

Exp.	Matching Technique	Baseline		SOCOM++ Trial 2 - Adjust Resource Constraint	
		St. Dev.	Mean	St. Dev.	Mean

1	1 NameAndPropertyAlignment	0.1014	0.9374	0.0922	0.9560
	2 StrucSubsDistAlignment	0.2505	0.7505	0.2379	0.7752
	3 ClassStructAlignment	0.2505	0.7505	0.0404	0.9791
	5 SMOANameAlignment	0.0582	0.9649	0.1633	0.9510
	6 SubsDistNameAlignment	0.1618	0.9041	0.1040	0.9431
	7 EditDistNameAlignment	0.0123	0.9909	0.0150	0.9914
	Avg.	**0.1391**	**0.8830**	**0.1088**	**0.9326**
2	1 NameAndPropertyAlignment	0.0909	0.9674	0.1483	0.9323
	2 StrucSubsDistAlignment	0.1509	0.9059	0.2188	0.8295
	3 ClassStructAlignment	0.1545	0.9440	0.1237	0.9356
	5 SMOANameAlignment	0.1556	0.9431	0.1233	0.9376
	6 SubsDistNameAlignment	0.1541	0.9372	0.2299	0.8664
	7 EditDistNameAlignment	0.0179	0.9913	0.0173	0.9898
	Avg.	**0.1207**	**0.9481**	**0.1435**	**0.9152**

The key findings from the baseline system, SOCOM++ trial one and SOCOM++ trial two are presented in table 5-8.

Table 5-8. Key Findings of Baseline, SOCOM++ Trial One and Two

Evaluations		**Baseline**	**SOCOM++ Trial 1 (default configuration)**	**SOCOM++ Trial 2**
Exp.1	Precision	0.3793	0. 4155	0.4437
	Recall	0.5640	0.6488	0.6616
	F-Measure	0.3782	0.4654	0.4674
	Confidence Level Mean	0.8830	0.9646	0.9326
	Confidence Level St.Dev.	0.1391	0.0613	0.1088
Exp.2	Precision	0.6918	0.7394	0.7569
	Recall	0.6057	0.6261	0.6521
	F-Measure	0.6347	0.6684	0.6886
	Confidence Level Mean	0.9481	0.9571	0.9152
	Confidence Level St.Dev.	0.1207	0.1065	0.1435

Compared to trial one, improvement in trial two is not always evident (e.g. lower confidence level mean and higher standard deviation were found in experiment two using the trial two configuration). As the difference between trial one and two is the lack of synonyms, one might intuitively assume that matching quality from trial two should be worse than those found of trial one. However, the opposite is shown (e.g. increased precision, recall and f-measure in experiment two; and improvements on all aspects in experiment one). Though the candidate AOLT pool has been reduced in trial two (compared to trial one), the selected AOLT results are therefore more likely to be the exact labels used by the target ontology (see table 5-6). Consequently, a greater number of matches can be generated with confidence, which leads to increased precision, recall and f-measure. Since this is the case, one could then assume that matches generated without analysing the embedded semantics (i.e. comparisons between semantic surroundings) would lead to poor matching outcome. Whether this assumption is true or not is investigated in the next trial.

5.4.2.3. Trial Three - adjust Embedded Semantics

Trial three investigates how the CLOM outcome is affected when the semantic surroundings (i.e. embedded semantics) are not taken into account during the AOLT selection process. Section 5.4.2.3.1 discusses the configuration details of trial three, followed by the findings in section 5.4.2.3.2.

5.4.2.3.1. Trial Setup

An assumption which can be derived from the findings in trial two is: matches generated without analysing the embedded semantics (i.e. semantic surroundings) may be of poor quality, since quality was not poorly affected even when there was a lack of candidate AOLT results to select from so long the semantic surroundings were included. The validity of this assumption is examined in trial three. As discussed previously in section 5.3, the embedded semantics of the source ontology is modelled by the `entry` element with `key` attribute `sourceSurrounding`, and the embedded semantics of the target ontology is modelled by the `entry` element with `key` attribute `targetSurrounding` in the system properties. Both elements can be configured to either `true` or `false`. In trial three, the semantic surroundings are disabled when both elements are set to the value `false`[97]. This configuration effectively disregards the semantic surroundings of both source and target ontology during the AOLT process.

Trial three is similar to trial one in that there are six types of candidate AOLT results available to the AOLT selection process. However, different from trial one, the configuration of trial three does not allow translation collisions to be resolved by comparisons made to semantic surroundings of ontological resources (since semantic surroundings are not accounted in the trial three configuration). In trial three, when a collision is detected between two entities E_1 and E_2, their candidate types are checked first as summarised in table 5-9. The entity with a lower type keeps the collided term as its AOLT result, and the other entity must seek an alternative translation. This is already demonstrated previously in trial one (see table 5-3, scenario i to x). When entity E_1 and E_2 are both of an equal candidate type, different from trial one however, the latter entity (one that is being considered by the AOLT selection process) will by default search for an alternative - without comparing to the source label's semantic

[97] Note that the execution constraint, i.e. `<entry key="default"/>` must be set to `false`, in order for the system to account settings of other properties (shown in figure 5-2).

surrounding (as shown in table 5-9, scenario xi). Alternative translations are achieved either by searching for a candidate AOLT with a higher type other than the current one (that is causing collision) or by attaching an integer (that is free of collision) to the collided term in the absence of any alternatives.

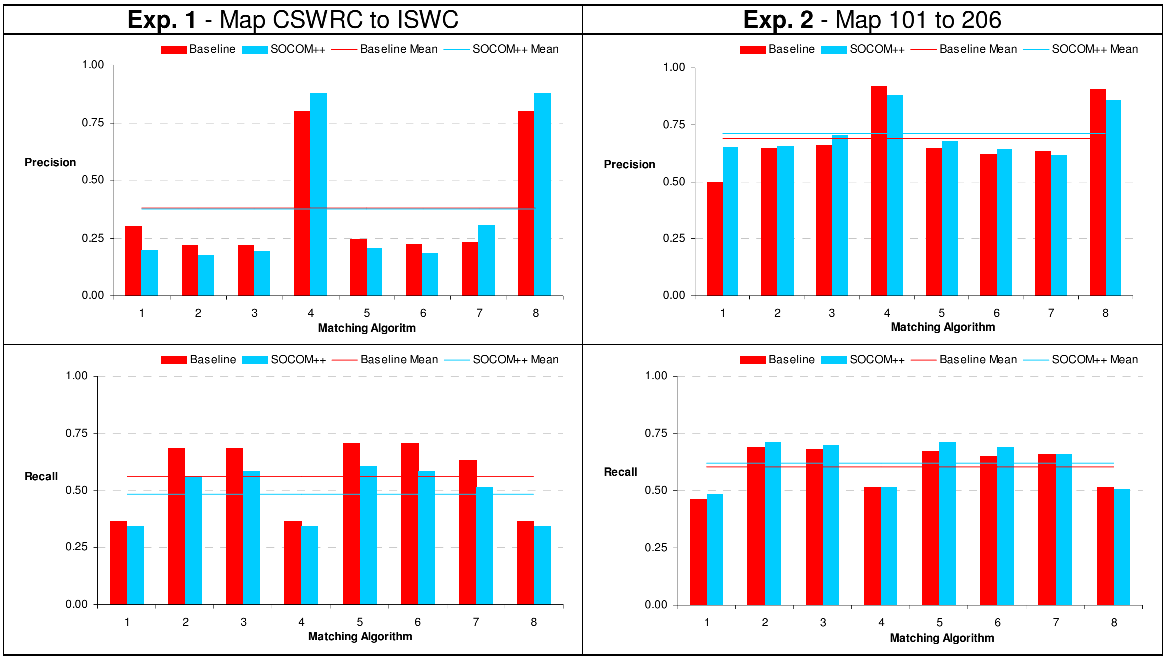
Table 5-9. Collision Resolution in Trial Three

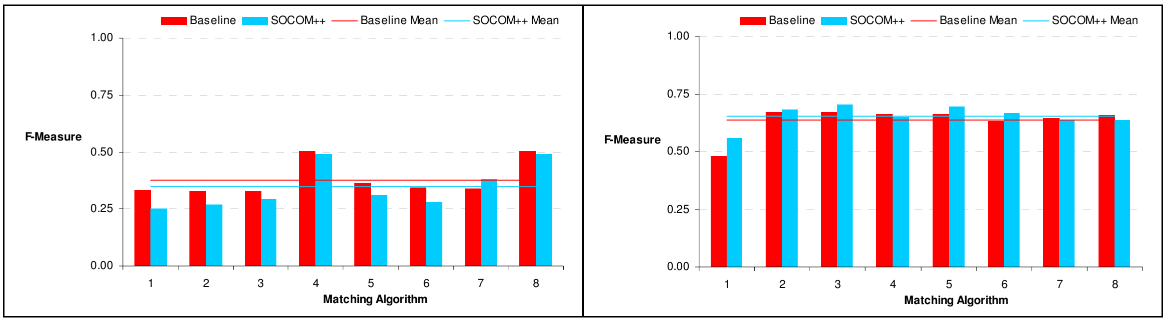
Collision Scenario	Candidate AOLT		Solution
	E_1	E_2	
i	type = 1	type = 2, 3, 4, 5 or 6	E_1 keeps the collided AOLT; E_2 seeks alternative AOLT with lowest possible type other than the current type.
ii	type = 2	type = 3, 4, 5 or 6	
iii	type = 3	type = 4, 5 or 6	
iv	type = 4	type = 5 or 6	
v	type = 5	type = 6	
vi	type = 2, 3, 4, 5 or 6	type = 1	E_2 keeps the collided AOLT; E_1 seeks alternative AOLT with the lowest possible type other than the current type.
vii	type = 3, 4, 5 or 6	type = 2	
viii	type = 4, 5 or 6	type = 3	
ix	type = 5 or 6	type = 4	
x	type = 6	type = 5	
xi	E_1 type = E_2 type		The existing entity that is already stored in the AOLT selection keeps the collided AOLT; the other entity seeks an alternative.

This trial configuration of SOCOM++ discussed in this section is applied to the two CLOM experiments outlined in section 5.4.1, findings are discussed next.

5.4.2.3.2. Findings and Analysis

The precision, recall and f-measure found in trial three for the two experiments are presented in figure 5-14. Findings from experiment one are shown in the left column and findings from experiment two are shown in the right column.





Legend: 1	NameAndPropertyAlignment	5	SMOANameAlignment
2	StrucSubsDistAlignment	6	SubsDistNameAlignment
3	ClassStructAlignment	7	EditDistNameAlignment
4	NameEqAlignment	8	StringDistAlignment

Figure 5-14. Precision, Recall, F-Measure in Trial Three

In experiment one, improvements in precision can only be seen in three matching algorithms: the *NameEqAlignment* algorithm, the *EditDistNameAlignment* algorithm and the *StringDistAlignment* algorithm. The number of correct matches generated by the majority of algorithms (i.e. the *NameAndPropertyAlignment* algorithm, the *StrucSubsDistAlignment* algorithm, the *ClassStructAlignment* algorithm, the *SMOANameAlignment* algorithm and the *SubsDistNameAlignment* algorithm) has not been improved when using the SOCOM++ trial three configuration. On average, a precision of 0.3769 was found in $M_{P2\text{-}T3}$, which is a 0.63% decline in the number of correct matches compared to M_B (at 0.3793). This deterioration is even more evident in recall, where no improvement is shown in any matching algorithm using the SOCOM++ trial three configuration. At an average recall of 0.4848, this is a fall by 14.04% in $M_{P2\text{-}T3}$ compared to M_B (at 0.5640). Consequently, the f-measure generated in $M_{P2\text{-}T3}$ is poorer in this trial than in M_B. On average, an f-measure of 0.3457 was found in $M_{P2\text{-}T3}$, which is an 8.59% of decrease compared to M_B (at 0.3782). The p-value from paired t-test on the f-measure scores is 0.05, which rejects the null hypothesis and suggests that there is a difference between the baseline and the SOCOM++ trial three configuration.

In experiment two, with the exception of the *NameEqAlignment* algorithm, the *EditDistNameAlignment* algorithm and the *StringDistAlignment* algorithm, all other algorithms generated higher precision in $M_{P2\text{-}T3}'$ than in M_B'. An average precision of 0.7105 was found in SOCOM++ in this trial, which is a 2.70% improvement from the baseline system (at 0.6918). $M_{P2\text{-}T3}'$ generated equal (in the case of the *NameEqAlignment* algorithm and the *EditDistNameAlignment* algorithm) or higher (in the case of the *NameAndPropertyAlignment* algorithm, the *StrucSubsDistAlignment* algorithm, the *ClassStructAlignment* algorithm, the *SMOANameAlignment* algorithm and the *SubsDistNameAlignment* algorithm) recall scores when using SOCOM++ with the exception of the *StringDistAlignment* algorithm. An average recall of 0.6224 was found in $M_{P2\text{-}T3}'$, which is a 2.76% improvement compared to M_B' (at 0.6057). Most algorithms generated higher f-measure scores in $M_{P2\text{-}T3}'$ in this trial except the *NameEq-Alignment* algorithm, the *EditDistNameAlignment* algorithm and the *StringDist-Alignment* algorithm. On average, an f-measure of 0.6529 was found in $M_{P2\text{-}T3}'$, which is an improvement of 2.87% compared to M_B' (at 0.6347). The average precision, recall and f-measure scores in $M_{P2\text{-}T3}'$ are higher than those found in M_B' in this trial, which may suggest improved quality in $M_{P2\text{-}T3}'$. However, this is not supported by the paired t-test carried out on the f-measure collected in $M_{P2\text{-}T3}'$ and M_B'. At a p-value of 0.148, the null hypothesis cannot be rejected. This finding suggests that although an improvement was noted in f-measure, the difference between them is not statistically significant. It is therefore difficult to argue that there has been an improvement on the matching quality when the SOCOM++ configuration was used in this trial.

The results from evaluating the confidence levels can be seen in table 5-10. Scatter plots generated using this data is shown in appendix E, section E.3, figure E-3. In experiment one, the average confidence mean is 0.8735 in $M_{P2\text{-}T3}$, which is a 1.08% decrease compared to M_B (at 0.8830). The average standard deviation in $M_{P2\text{-}T3}$ is 0.1540, which is a 10.71% increase compared to M_B (at 0.1391). This finding suggests that there has not been an improvement in the matches' confidence levels in SOCOM++ in this trial. A similar result is found in experiment two. The average confidence mean is decreased by 1.70% in $M_{P2\text{-}T3}'$ to 0.9320 compared to M_B' (at 0.9481). The standard deviation is increased by 8.04% to 0.1304 in $M_{P2\text{-}T3}'$ compared to M_B' (at 0.1207).

Table 5-10. Confidence Data from Trial Three

Exp.	Matching Technique	Baseline		SOCOM++ Trial 3 - Adjust Embedded Semantics	
		St.Dev.	Mean	St.Dev.	Mean
i	1 NameAndPropertyAlignment	0.1014	0.9374	0.1471	0.8897
	2 StrucSubsDistAlignment	0.2505	0.7505	0.2125	0.6771
	3 ClassStructAlignment	0.2505	0.7505	0.1841	0.9207
	5 SMOANameAlignment	0.0582	0.9649	0.1886	0.9138
	6 SubsDistNameAlignment	0.1618	0.9041	0.1758	0.8536
	7 EditDistNameAlignment	0.0123	0.9909	0.0158	0.9859
	Avg.	**0.1391**	**0.8830**	**0.1540**	**0.8735**
ii	1 NameAndPropertyAlignment	0.0909	0.9674	0.1358	0.9377
	2 StrucSubsDistAlignment	0.1509	0.9059	0.1861	0.8726
	3 ClassStructAlignment	0.1545	0.9440	0.1163	0.9499
	5 SMOANameAlignment	0.1556	0.9431	0.1170	0.9476
	6 SubsDistNameAlignment	0.1541	0.9372	0.2143	0.8900
	7 EditDistNameAlignment	0.0179	0.9913	0.0131	0.9939
	Avg.	**0.1207**	**0.9481**	**0.1304**	**0.9320**

In summary, this trial shows a much less superior performance of SOCOM++, as predicated in the assumption previously. Particularly when dealing with ontologies containing natural language pairs from different language families (i.e. experiment one), the trial three configuration of SOCOM++ proves to be far from desired. Not only are the precision, recall and f-measure not been improved, but the matches are less confident also with more dispersed confidence levels. Table 5-11 presents the key findings from baseline, and SOCOM++ trial one, two and three. Trial three achieved the worst matching quality (lower values in precision, recall, f-measure and mean confidence level, and higher values in confidence level standard deviations) in both experiments compared to the previous two trials (where both trial configurations accounted semantic surroundings during the AOLT selection). This finding shows that semantic surrounding is an essential input for the AOLT process, even when a small AOLT candidate pool is available. This validates the assumption at the start of this trial.

Table 5-11. Key Findings of Baseline, SOCOM++ Trial One, Two and Three

Evaluations		Baseline	SOCOM++ Trial 1 (default configuration)	SOCOM++ Trial 2	SOCOM++ Trial 3
Exp.1	Precision	0.3793	0. 4155	0.4437	0.3769
	Recall	0.5640	0.6488	0.6616	0.4848
	F-Measure	0.3782	0.4654	0.4674	0.3457
	Confidence Level Mean	0.8830	0.9646	0.9326	0.8735
	Confidence Level St.Dev.	0.1391	0.0613	0.1088	0.1540
Exp.2	Precision	0.6918	0.7394	0.7569	0.7105
	Recall	0.6057	0.6261	0.6521	0.6224
	F-Measure	0.6347	0.6684	0.6886	0.6529
	Confidence Level Mean	0.9481	0.9571	0.9152	0.9320
	Confidence Level St.Dev.	0.1207	0.1065	0.1435	0.1304

5.4.3. Three Trials to execute a Second Iteration of the AOLT Process

This section presents another three trials of SOCOM++ that focus on carrying out a second iteration of the AOLT process (in contrast to the previous three trials discussed in section 5.4.2). Trial four (discussed in section 5.4.3.1), five (discussed in section 5.4.3.2) and six (discussed in section 5.4.3.3) each presents and investigates a different selection rationale for the AOLT process during its second iteration.

5.4.3.1. Trial Four - adjust Task Intent: Optimising Correctness

Trial four investigates how the optimising correctness task intent may impact on the matching quality generated using SOCOM++. Section 5.4.3.1.1 discusses the configuration details, and section 5.4.3.1.2 presents the findings and analysis.

5.4.3.1.1. Trial Setup

Optimising correctness aims to generate as many precise matches as possible in the second iteration of SOCOM++, by applying a strict cut-off point to the matches generated from the first iteration. As discussed in section 5.3, task intent is modelled by the `entry` element with the `key` attribute `correctnessOptimise` and the `entry` element with the `key` attribute `completenessOptimise`. Both can be configured to `true` (i.e. enabled) or `false` (i.e. disabled), but only one can be enabled at a time. In trial four, optimising correctness is enabled, and two iterations of SOCOM++ are executed. In the first iteration, the default AOLT selection (i.e. trial one - discussed in section 5.4.2.1, also see the example shown in figure 5-9) is executed to generate an initial set of matches using a specific MOM algorithm. The system then assumes that the matches with 1.0 confidence levels are most likely to be correct and computes the selection rationale (i.e. how the AOLT results are derived) behind them. An example of the analysis generated through this process in the XML format is presented in figure 5-15. The DTD for this output can be found in appendix D, section D.3, figure D-8.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TaskIntent SYSTEM "TaskIntent.dtd">
<TaskIntent algorithm="SMOANameAlignment" intent="correctnessOptimise" matches="119.0"
estimate="32.0">
    <Entry count="16.0" media="both" type="1" usage="0.5"/>
    <Entry count="8.0" media="google" type="1" usage="0.25"/>
    <Entry count="3.0" media="bing" type="1" usage="0.09375"/>
    <Entry count="2.0" media="BHT" type="4" usage="0.0625"/>
    <Entry count="1.0" media="BHT" type="2" usage="0.03125"/>
    <Entry count="1.0" media="google" type="6" usage="0.03125"/>
    <Entry count="1.0" media="bing" type="6" usage="0.03125"/>
</TaskIntent>
```


Figure 5-15. An Example Output from Task Intent Analysis - Optimising Correctness

In the example shown in figure 5-15, the analysis is computed for the *SMOANameAlignment* algorithm (stored in the attribute `algorithm` of the root element `TaskIntent`) with the intent of optimising correctness (stored in the attribute `method` of the root element) in the matches generated. After the first interaction of the system (i.e. applying the default configuration when selecting AOLT results), a total of 119 matches (stored in the attribute `matches` of the root element) were generated by the *SMOA-NameAlignment* algorithm. Among which, 32 of them (stored in the attribute `estimate` of the root element) had confidence levels of 1.0. The rationale behind these 32 "correct" matches is stored as attribute values in the child element: `Entry`. In the example, 16 (stored in the attribute `count` of the first `Entry` element) of those "correct" matches were generated using AOLT results that were of type 1 (stored in the attribute `type` of the `Entry` element) and had been agreed by both MT tools (stored in the attribute `media` of the `Entry` element), which yields a usage of 50% (stored in the attribute `usage` of the `Entry` element, calculated as count/estimate). Similarly, usages are calculated for all combinations (i.e. combination of `type` and `media`) that appeared in the "correct" matches for each matching algorithm as shown in this example.

In the second iteration, the rationales generated from the first iteration are treated as a ranked list of AOLT selection strategies. Note that the order of the AOLT selection strategies will differ depending on the matching algorithm applied, because the ranked lists are generated on a per-MOM-algorithm basis. In the example shown in figure 5-15, when using the *SMOANameAlignment* algorithm in the second iteration, the candidate AOLT results (which are stored in the AOLT record, see section 5.3, figure 5-8 for an example) with `type="1"` and `media="both"` are most preferred translations for the source labels. If such candidates are unavailable, in second place, the AOLT results with `type="1"` and `media="google"` will be selected. In the absence of the above, in third place, the AOLT results with `type="1"` and `media="bing"` will be selected and so on. When several AOLT selection strategies acquire equal usages, for example in figure 5-15, the last three `Entry` elements all obtained the same usage of 0.03125, in such situation, any one of these selection techniques is considered suitable, as long as no collision is caused. This is discussed next.

The AOLT selection process discussed thus far is repeated for each MOM matching algorithm, and the selection strategies are applied accordingly in the second iteration (as mentioned earlier, the `type` and `media` combination as well as the order of them vary depending on the MOM algorithm). When collisions are detected, the system checks the origins of the collided term and prioritises the resource with higher ranked selection strategy (i.e. one that scored a higher usage in the task intent analysis) where possible. A summary of the resolutions is presented in table 5-12. Given a pair of entities E_1 and E_2, the entity with the AOLT result that derived from a higher selection strategy will keep the collided term, and the other entity must seek an alternative AOLT with a lower selection strategy from the AOLT record, as demonstrated by scenario i and ii in table 5-12. When both entities choose the same AOLT result with equal rank, the system checks whether alternative AOLT results exist for each of them. If alternative AOLT results are only available for one entity, then this entity must seek an alternative whereas the other entity keeps the collided term, as shown in scenario iii. If alternative AOLT results exist for both entities, then the second entity (i.e. one that came after the collided term has already been stored as an AOLT for an earlier entity) will seek alternative while the first entity keeps the collided term, as shown in scenario

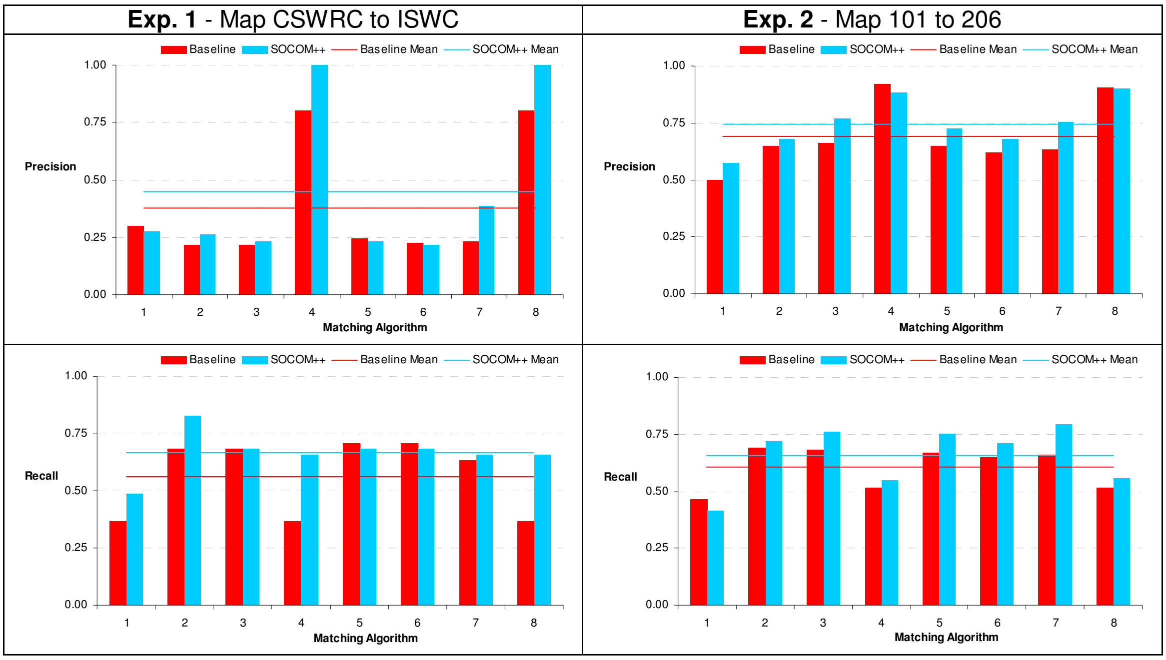
iv. When collisions cannot be solved using solutions presented in table 5-12 (e.g. alternative AOLT results simply do not exist in the desired `type` and `media` combination), the system retreats to the default resolution technique used in trial one (discussed previously in section 5.4.2.1.1). This SOCOM++ trial configuration is evaluated next using experiments outlined in section 5.4.1.

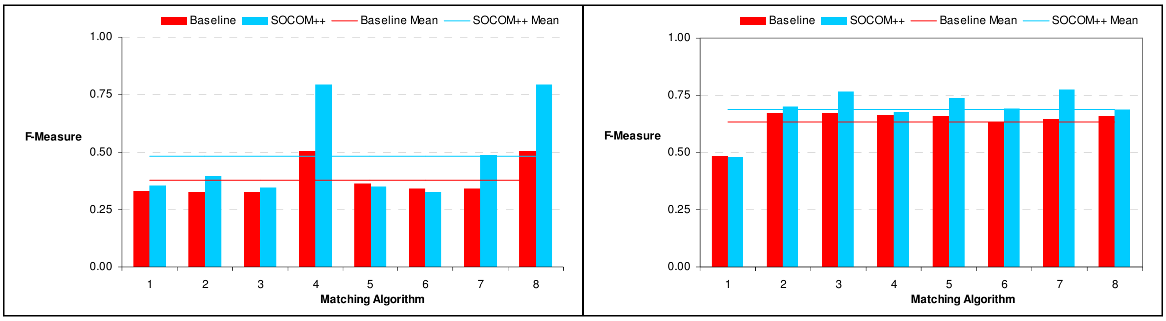
Table 5-12. Collisions Resolution in Trial Four

Collision Scenario	Candidate AOLT		Solution
	E_1	E_2	
i	Higher rank in TaskIntent.xml	Lower rank in TaskIntent.xml	E_1 keeps the collided AOLT; E_2 seeks alternative AOLT with lower ranked selection strategy.
ii	Lower rank in TaskIntent.xml	Higher rank in TaskIntent.xml	E_2 keeps the collided AOLT; E_1 seeks alternative AOLT with lower ranked selection strategy.
iii	Equal rank in TaskIntent.xml, one entity has alternative candidate AOLT results, the other entity has no alternative candidate AOLT.		The entity with no alternative AOLT keeps the collided AOLT; the other entity seeks alternative AOLT with lower ranked selection strategy.
iv	Equal rank in TaskIntent.xml, both entities have alternative candidate AOLT results.		The first entity keeps the collided AOLT; the second entity seeks alternative AOLT with lower ranked selection strategy.

5.4.3.1.2. Findings and Analysis

The precision, recall and f-measure generated in trial four are shown in figure 5-16. The results from experiment one are presented in the left column. The results from experiment two are presented in the right column.





Matching Algorithms:	1	NameAndPropertyAlignment	5	SMOANameAlignment
	2	StrucSubsDistAlignment	6	SubsDistNameAlignment
	3	ClassStructAlignment	7	EditDistNameAlignment
	4	NameEqAlignment	8	StringDistAlignment

Figure 5-16. Precision, Recall, F-Measure in Trial Four

In experiment one, with the exception of the *SMOANameAlignment* algorithm and the *SubsDistNameAlignment* algorithm, all other algorithms achieved higher precision in $M_{P2\text{-}T4}$. The improvement is particularly evident in the case of the *NameEq-Alignment* algorithm and the *StringDistAlignment* algorithm, where a precision score of 1.0 had been achieved. This is the highest precision any algorithm was able to obtain in the trials so far. On average, a precision of 0.4497 was generated in $M_{P2\text{-}T4}$, which is an 18.56% improvement compared to M_B (at 0.3793). This average precision is the highest score in all trials carried out so far. As more correct matches are generated, the recall scores are thus increased at the same time[98]. Similar results can be seen in the recall scores generated. On average, a recall of 0.6677 was found in $M_{P2\text{-}T4}$, which is an 18.39% improvement of the M_B (at 0.5640). Overall, an average f-measure of 0.4800 was found in $M_{P2\text{-}T4}$, which is an improvement by 26.92% compared to M_B (at 0.3782). However, the p-value generated from paired t-test yields 0.06, which suggests that there is not enough evidence to conclude a difference between the two systems in this trial, though the average f-measure may suggest otherwise. Nevertheless, the goal of this trial - optimising the correctness of matches generated in the second iteration - has been achieved as shown through the highest precision score achieved by SOCOM++ to date.

In experiment two, optimising correctness is less evident in comparison to experiment one. Particularly in the case of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm, decreases of precision scores have been found. On average, a precision of 0.7449 was found in $M_{P2\text{-}T4}'$, which is an improvement of 7.68% compared to M_B' (at 0.6918). This is not the highest precision that has been achieved in this experiment (see section 5.4.2.2.2 trial two). Except the *NameAndProperty-*

[98] Precision = N/X; Recall = N/R where X is the total number of matches found, N is the correct matches among X, and R is the gold standard. While N increases and R remains static, Recall is thus increased as a result.

Alignment algorithm, recall is improved for all other algorithms in $M_{P2\text{-}T4}$'. At an average of 0.6572, this is an 8.50% improvement of M_B' (at 0.6057). Overall, an average f-measure of 0.6892 was found in $M_{P2\text{-}T4}$', which is an improvement by 8.59% on M_B' (at0.6347). The p-value generated from paired t-test carried on the f-measure scores is 0.01, suggesting the statistical significance of the findings in this experiment.

Table 5-13 presents the evaluation results of the confidence levels from the two experiments. Scatter plots generated using this data shown can be found in appendix E, section E.4, figure E-4. In experiment one, matches in $M_{P2\text{-}T4}$ are more confident with less dispersed confidence levels. An average confidence mean of 0.9472 was found in $M_{P2\text{-}T4}$, which is an improvement by 7.27% compared to M_B (at 0.8830). An average standard deviation of 0.0832 was found in $M_{P2\text{-}T4}$, which is a 40.19% improvement from M_B (at 0.1391). In contrast, the evaluation results found from experiment two are less positive. The matches in $M_{P2\text{-}T4}$' are less confident (i.e. lower mean confidence level), however their confidence levels are less dispersed (i.e. lower standard deviation) compared to M_B'. An average mean of 0.9436 was found in $M_{P2\text{-}T4}$', which is a decrease by 0.47% compared to M_B' (at 0.9481). An average standard deviation of 0.1182 was found in $M_{P2\text{-}T4}$', which is an improvement by 2.07% compared to M_B' (at 0.1207).

Table 5-13. Confidence Data from Trial Four

Exp.	Matching Technique	Baseline		SOCOM++ Trial 4 – Adjust Task Intent (Optimising Correctness)	
		St.Dev.	Mean	St.Dev.	Mean
i	1 NameAndPropertyAlignment	0.1014	0.9374	0.0615	0.9830
	2 StrucSubsDistAlignment	0.2505	0.7505	0.2472	0.7479
	3 ClassStructAlignment	0.2505	0.7505	0.0390	0.9900
	5 SMOANameAlignment	0.0582	0.9649	0.0390	0.9900
	6 SubsDistNameAlignment	0.1618	0.9041	0.1083	0.9730
	7 EditDistNameAlignment	0.0123	0.9909	0.0040	0.9992
	Avg.	**0.1391**	**0.8830**	**0.0832**	**0.9472**
ii	1 NameAndPropertyAlignment	0.0909	0.9674	0.1166	0.9598
	2 StrucSubsDistAlignment	0.1509	0.9059	0.1816	0.8904
	3 ClassStructAlignment	0.1545	0.9440	0.1050	0.9532
	5 SMOANameAlignment	0.1556	0.9431	0.1048	0.9548
	6 SubsDistNameAlignment	0.1541	0.9372	0.1835	0.9132
	7 EditDistNameAlignment	0.0179	0.9913	0.0178	0.9903
	Avg.	**0.1207**	**0.9481**	**0.1182**	**0.9436**

Table 5-14 presents the key findings from the baseline system, SOCOM++ trial one (default configuration) and SOCOM++ trial four. As trial four is essentially the default configuration added with a second iteration (that is enabled by the optimising correctness task intent), it is thus of interest to compare trial four to trial one (as opposed to trial two or three). In summary, correct matches generated in the second iteration are shown to be greater than those generated in the first iteration of the system (see higher precision, recall and f-measure values from both experiments in trial four compared to trial one). Although there is a trade-off on matches' confidence levels - in both experiments, lower confidence level means and higher standard deviations were found. Nevertheless, the trial four configuration did improve the matching quality in terms of precision, recall and f-measure, which was the goal of this trial setup. Motivated by this result, the possibility of optimising completeness in task intent is investigated and discussed next.

Table 5-14. Key Findings of Baseline, SOCOM++ Trial One and Four

Evaluations		Baseline	SOCOM++ Trial 1 (default configuration)	SOCOM++ Trial 4
Exp.1	Precision	0.3793	0. 4155	0.4497
	Recall	0.5640	0.6488	0.6677
	F-Measure	0.3782	0.4654	0.4800
	Confidence Level Mean	0.8830	0.9646	0.9472
	Confidence Level St.Dev.	0.1391	0.0613	0.0832
Exp.2	Precision	0.6918	0.7394	0.7449
	Recall	0.6057	0.6261	0.6572
	F-Measure	0.6347	0.6684	0.6892
	Confidence Level Mean	0.9481	0.9571	0.9436
	Confidence Level St.Dev.	0.1207	0.1065	0.1182

5.4.3.2. Trial Five - Optimising Completeness

Trial five investigates how the configuration with the task intent of optimising completeness affects the mapping outcome. Section 5.4.3.2.1 presents the configuration details of this trial. Section 5.4.3.2.2 presents the findings and analysis.

5.4.3.2.1. Trial Setup

When optimising completeness is enabled in task intent (by setting the `entry` element with attribute `completenessOptimise` to `true`), similarly to optimising correctness, two iterations of the AOLT process are executed. However, different from optimising correctness, the system assumes that all matches (with any confidence levels) are correct for a specific MOM algorithm applied. Through this assumption, the system does not discard any match generated from the first iteration pre-maturely (i.e. a match may still be correct even though it has lower than 1.0 confidence level), and ensures as many matches as possible are analysed in an effort to optimise the completeness of correct matches generated in the second iteration. An example output is shown in figure 5-17. Its DTD can be found in appendix D, section D.3, figure D-8.


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE TaskIntent SYSTEM "TaskIntent.dtd">
<TaskIntent algorithm="SMOANameAlignment" intent="completenessOptimise" matches="119.0"
estimate="119.0">
    <Entry count="24.0" media="google" type="6" usage="0.20168067226890757"/>
    <Entry count="20.0" media="both" type="5" usage="0.16806722689075632"/>
    <Entry count="17.0" media="BHT" type="4" usage="0.14285714285714285"/>
    <Entry count="16.0" media="both" type="1" usage="0.13445378151260504"/>
```

```
    <Entry count="13.0" media="bing" type="6" usage="0.1092436974789916"/>
    <Entry count="9.0" media="google" type="1" usage="0.07563025210084033"/>
    <Entry count="6.0" media="both" type="3" usage="0.05042016806722689"/>
    <Entry count="6.0" media="google" type="3" usage="0.05042016806722689"/>
    <Entry count="3.0" media="bing" type="1" usage="0.025210084033613446"/>
    <Entry count="3.0" media="external" type="external" usage="0.025210084033613446"/>
    <Entry count="2.0" media="BHT" type="2" usage="0.01680672268907563"/>
</TaskIntent>
```


Figure 5-17. An Example Output from Task Intent Analysis - Optimising Completeness

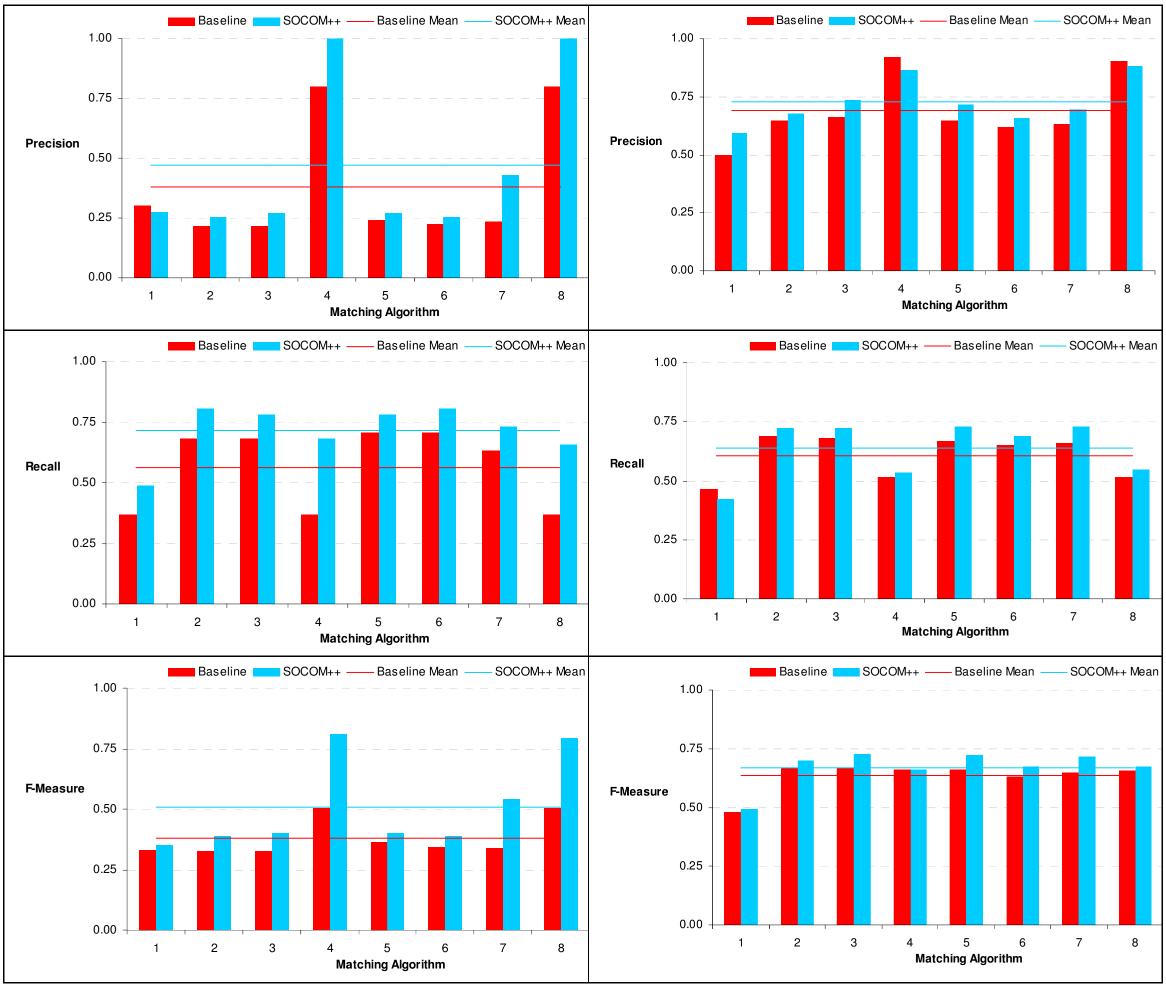
In the example shown in figure 5-17, the analysis is computed for the *SMOA-NameAlignment* algorithm (stored as attribute value: `algorithm` in the root element `TaskIntent`) with the intent of optimising completeness (stored as attribute value: `intent` in the root element). In the first iteration, 119 matches (stored as attribute value: `matches` in the root element) were generated, and all 119 of them (stored as attribute value: `estimate` in the root element) are estimated to be correct in the analysis. A list of rationales (stored as attribute values in the `Entry` elements) used to select the AOLT results for these "correct" matches are then calculated grouped by type and media. Ranked in first place with the highest usage, 24 matches used AOLT results of `type="6"` and `media="google"` in the first iteration. In second place, 20 matches used AOLT results of `type="5"` and `media="both"`, and so on. In the second iteration of SOCOM++, AOLT results are selected based on this ranking.

Similarly to trial four (discussed in section 5.4.3.1.1, see table 5-12), translation collisions are solved in the same way in trial five. Note that in the example shown in figure 5-17, 3 matches were of `type="external"` and `media="external"`. These are matches made between externally defined resources, e.g. `rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#List"` is an RDF vocabulary that is defined by the World Wide Web Consortium (W3C). Although categorised, such rationale cannot influence the selection of AOLT results during the second iteration of SOCOM++, because syntax specifications are not changed during the ontology rendition process in the first iteration (discussed in chapter 1, section 1.7). SOCOM++ trial five configuration is evaluated in the experiments outlined in section 5.4.1 next.

5.4.3.2.2. Findings and Analysis

The precision, recall and f-measure scores generated in trial five are presented in figure 5-18. The results from experiment one are presented in the left column. The results from experiment two are presented in the right column.

Exp. 1 – Map CSWRC to ISWC	**Exp. 2** – Map 101 to 206



Matching Algorithms: 1 NameAndPropertyAlignment 5 SMOANameAlignment
2 StrucSubsDistAlignment 6 SubsDistNameAlignment
3 ClassStructAlignment 7 EditDistNameAlignment
4 NameEqAlignment 8 StringDistAlignment

Figure 5-18. Precision, Recall, F-Measure in Trial Five

In experiment one, with the exception of the *NameAndPropertyAlignment* algorithm, precision scores of all other algorithms were improved in $M_{P2\text{-}T5}$. An average of 0.4696 was found in $M_{P2\text{-}T5}$, which is a 23.81% improvement compared to M_B (with at 0.3793). Significant improvement in the recall scores can be seen in all matching algorithms, particularly in the case of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm. An average recall of 0.7165 was found in $M_{P2\text{-}T5}$, which is an improvement by 27.04% compared to M_B (at 0.5640). This is the highest average recall score that has been achieved in this experiment by any trial so far. This finding shows that the optimising completeness configuration in trial four has been successful in this experiment. With improved precision and recall, the f-measure scores are consequently increased. An average of 0.5098 was found in $M_{P2\text{-}T5}$, which is an improvement by 34.80% compared to M_B (at 0.3782). The p-value generated from paired t-test carried out on the f-measure scores from the two systems yields 0.016, which supports the statistical significance of the findings so far. This further validates the improved matching quality in $M_{P2\text{-}T5}$.

In experiment two, with the exception of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm, all other algorithms generated higher precision in $M_{P2\text{-}T5}$'. An average precision of 0.7288 was found in $M_{P2\text{-}T5}$', which is an improvement by 5.35% compared to M_B' (at 0.6918). The recall for most matching algorithms (with the exception of the *NameAndPropertyAlignment* algorithm) has also been improved in $M_{P2\text{-}T5}$'. An average recall of 0.6379 was found in $M_{P2\text{-}T5}$', which is a 5.32% improvement from M_B' (at 0.6057). This is not the highest recall mean that was ever achieved in this experiment, as the average recall achieved in trial two and trial four are both higher. This finding suggests that the trial five configuration is not as suitable in experiment two as it is in experiment one. Overall, improvement in f-measure can be seen in all matching algorithms. An average f-measure of 0.6715 was found in $M_{P2\text{-}T5}$', which is an improvement by 5.80% compared to M_B' (at 0.6347). The p-value further supports the improvement in matching quality in $M_{P2\text{-}T5}$': at 0.004, it validates the statistical significance of the results above.

Table 5-15 presents the results from evaluating the confidence levels of the matches generated in both experiments. Scatter plots generated using this data can be found in appendix E, section E.5, figure E-5. In experiment one, an average confidence mean of 0.9252 and an average standard deviation of 0.0973 was found in $M_{P2\text{-}T5}$. This is an average increase by 4.78% on the confidence mean and a decrease by 30.05% on the standard deviation compared to M_B. This finding suggests that the matches generated using SOCOM++ were more confident with less dispersed confidence levels in this experiment. In experiment two, an average confidence mean of 0.9441 and an average standard deviation of 0.1205 was found in $M_{P2\text{-}T5}$'. This is an average 0.17% improvement on standard deviation, but a 0.42% decrease on confidence mean. This finding suggests that the matches generate by SOCOM++ may have less dispersed confidence levels, but their confidence means are not quite as high in this experiment.

In summary, this trial run has successfully demonstrated the optimising completeness feature when working with ontologies containing natural language pairs from different language families. However, this configuration was not as successful when dealing with ontologies containing natural language pairs from the same language family. Table 5-16 presents the key findings from the baseline system, SOCOM++ trial one (default configuration) and SOCOM++ trial five. The effectiveness of the trial five configuration is evident through the increased recall values generated in both experiments compared to the default configuration (i.e. trial one). However, there is a trade-off regarding the confidence levels: as shown in table 5-16, decreased confidence level means and increased standard deviations were found in both experiments.

Table 5-15. Confidence Data from Trial Five

Exp.	Matching Technique	Baseline		SOCOM++ Trial 5 - Adjust Task Intent (Optimising Completeness)	
		St.Dev.	Mean	St.Dev.	Mean
i	1 NameAndPropertyAlignment	0.1014	0.9374	0.0943	0.9597
	2 StrucSubsDistAlignment	0.2505	0.7505	0.2336	0.7355
	3 ClassStructAlignment	0.2505	0.7505	0.0507	0.9734
	5 SMOANameAlignment	0.0582	0.9649	0.0507	0.9734
	6 SubsDistNameAlignment	0.1618	0.9041	0.1405	0.9189
	7 EditDistNameAlignment	0.0123	0.9909	0.0141	0.9904
	Avg.	**0.1391**	**0.8830**	**0.0973**	**0.9252**
ii	1 NameAndPropertyAlignment	0.0909	0.9674	0.1079	0.9619
	2 StrucSubsDistAlignment	0.1509	0.9059	0.1600	0.9022
	3 ClassStructAlignment	0.1545	0.9440	0.1061	0.9525
	5 SMOANameAlignment	0.1556	0.9431	0.1498	0.9422
	6 SubsDistNameAlignment	0.1541	0.9372	0.1815	0.9151
	7 EditDistNameAlignment	0.0179	0.9913	0.0177	0.9905
	Avg.	**0.1207**	**0.9481**	**0.1205**	**0.9441**

Table 5-16. Key Findings of Baseline, SOCOM++ Trial One and Five

Evaluations		Baseline	SOCOM++ Trial 1 (default configuration)	SOCOM++ Trial 5
Exp.1	Precision	0.3793	0. 4155	0.4696
	Recall	0.5640	0.6488	0.7165
	F-Measure	0.3782	0.4654	0.5098
	Confidence Level Mean	0.8830	0.9646	0.9252
	Confidence Level St.Dev.	0.1391	0.0613	0.0973
Exp.2	Precision	0.6918	0.7394	0.7288
	Recall	0.6057	0.6261	0.6379
	F-Measure	0.6347	0.6684	0.6715
	Confidence Level Mean	0.9481	0.9571	0.9441
	Confidence Level St.Dev.	0.1207	0.1065	0.1205

Optimising correctness (trial four) and optimising completeness (trial five) can be thought of as two extremes when assessing matches generated in the first iteration of SOCOM++, where the former applies a highest possible cut-off point (i.e. only matches with 1.0 confidence levels are assumed to be correct) and the latter applies a lowest possible cut-off point (i.e. assume all matches generated in the first iteration are correct). Clearly, there can be many other cut-off points between these two extremes. For example, 0.5 is a natural cut-off point between the value 0.0 and 1.0, whereby equal or greater than 0.5 indicates an incline towards confident, and less than 0.5 indicates an incline towards not confident. Although any value between 0.0 and 1.0 is acceptable, 0.5 is most interesting as it is a natural division point between the two extremes. This is implemented in trial six, discussed next.

5.4.3.3. Trial Six - adjust Pseudo Feedback

Trial six focuses on the configuration of the pseudo feedback property, which offers the user with flexible cut-off points (any value between 0.0 and 1.0) when assessing matches for further iterations of SOCOM++. In particular, a cut-off point of 0.5 is investigated in this trial. Section 5.4.3.3.1 presents the configuration details, followed by experimental findings and analysis in section 5.4.3.3.2.

5.4.3.3.1. Trial Setup

As discussed in section 5.3, the pseudo feedback property is modelled by setting a cut-off point for the assessment of matches generated in the first iteration of SOCOM++. This is achieved by setting the `entry` element with the `key` attribute `threshold` to any value that is between 0.0 and 1.0 (see figure 5-2). This value is then treated as the cut-off point for confidence levels.

In this trial, the pseudo feedback property is configured as `<entry key= "threshold">0.5</entry>`. Instead of assuming all matches are correct (as shown in trial five), or only matches with 1.0 confidence levels are correct (as shown in trial four), this configuration assumes any match with confidence level that is equal to or above 0.5 is correct for a specific MOM algorithm. As confidence levels range between 0.0 and 1.0, 0.5 is a natural division point where matches would either incline towards being either confident (i.e. equal or above 0.5) or not confident (i.e. below 0.5). Based on this assumption, a set of AOLT selection rationale is computed. Note that trial six does not attempt to present an exhaustive list of all possible cut-off points (since it can be anything between the value of 0.0 and 1.0), or aim to establish the best possible cut-off point for the two experiments (as that will require extensive tests on various cut-off points which will lead to an exhaustive list). It is simply an example of configurable cut-off points that is offered by the pseudo feedback feature. Similar to what was discussed in trial four and five, selection rationales are generated on a per-matching-algorithm basis. An example output from the pseudo feedback analysis is shown in figure 5-19. Its DTD can be found in appendix D, section D.3, figure D-9.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PseudoFeedback SYSTEM "PseudoFeedback.dtd">
<PseudoFeedback    algorithm="SMOANameAlignment"    threshold="0.5"    matches="119.0"
estimate="60.0">
    <Entry count="16.0" media="both" type="1" usage="0.266666667"/>
    <Entry count="10.0" media="google" type="6" usage="0.166666667"/>
    <Entry count="9.0" media="google" type="1" usage="0.15"/>
    <Entry count="8.0" media="bing" type="6" usage="0.133333333"/>
    <Entry count="6.0" media="both" type="5" usage="0.1"/>
    <Entry count="4.0" media="BHT" type="4" usage="0.0666666667"/>
```

```
    <Entry count="3.0" media="bing" type="1" usage="0.05"/>
    <Entry count="2.0" media="BHT" type="2" usage="0.0333333333"/>
    <Entry count="1.0" media="google" type="3" usage="0.0166666667"/>
    <Entry count="1.0" media="both" type="3" usage="0.0166666667"/>
</PseudoFeedback>
```

Figure 5-19. An Example Ouput from Pseudo Feedback Analysis

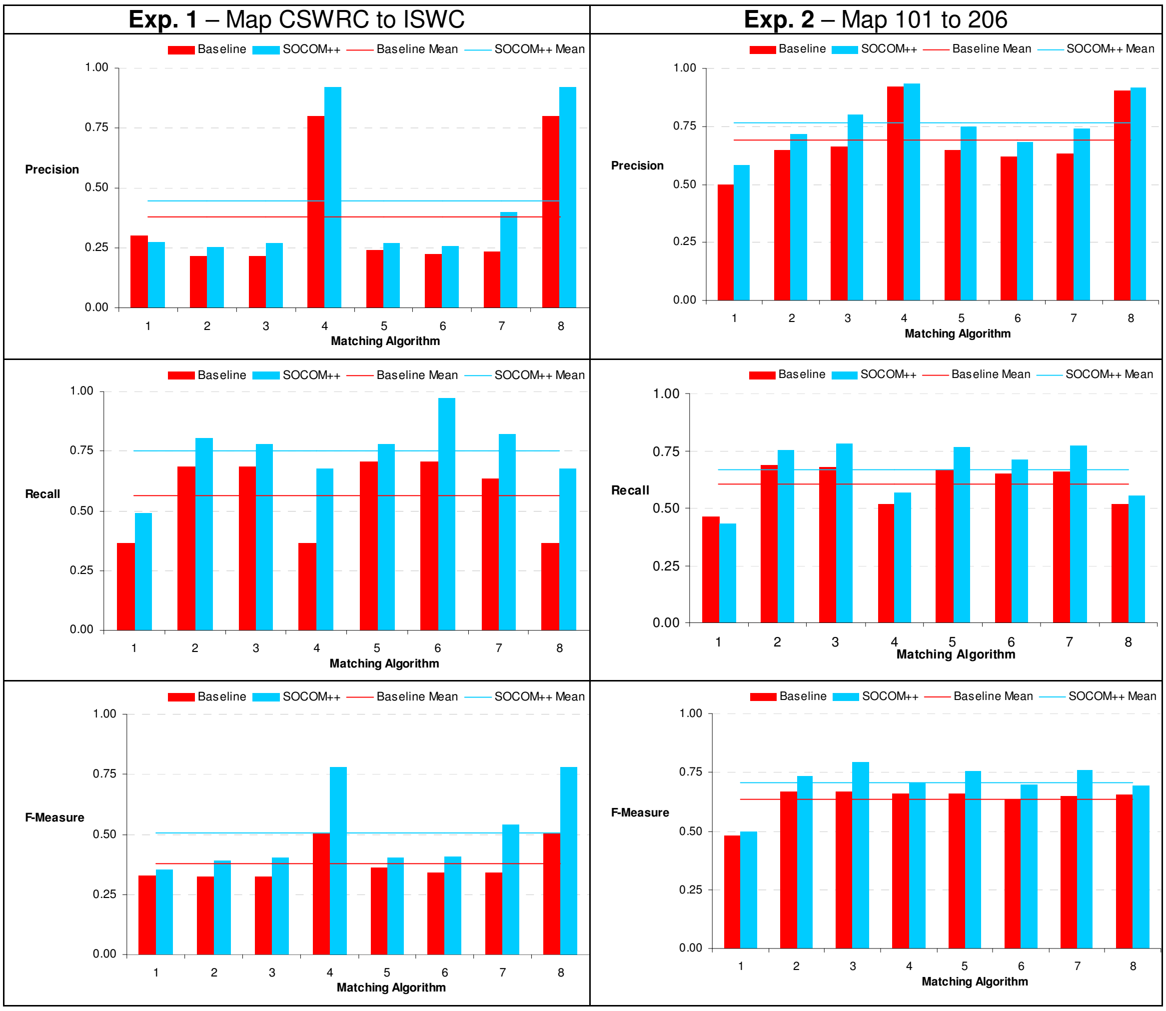
The example shown in figure 5-19 is generated for the *SMOANameAlignment* algorithm (stored as attribute value: `algorithm` in the root element `PseudoFeedback`) when the threshold is 0.5 (stored as attribute value: `threshold` of the root element). A total of 119 matches (stored as attribute value: `matches` in the root element) were generated in the first iteration, 60 of which are estimated to be correct (stored as attribute value: `estimate` in the root element) using the threshold. Ranked in first place, the most often used AOLT results are of `type="1"` and `media="both"` (see the first child element) among the "correct" matches, followed by several other selection strategies. Note how the combinations and the rankings of them differ from figure 5-15 and figure 5-17 for the same matching algorithm. In the second iteration of the system, the AOLT results are selected with preferences to the ranked list shown in figure 5-19 for the *SMOANameAlignment* algorithm. Translation collisions are solved in the same fashion as in trial four and five (discussed in section 5.4.3.1.1 and section 5.4.3.2.1 respectively). The evaluation of this SOCOM++ trial configuration is discussed next using the experiments outlined in section 5.4.1.

5.4.3.3.2. Findings and Analysis

The precision, recall and f-measure scores generated in trial five are presented in figure 5-20. The results from experiment are shown in the left column, and the results from experiment two are shown in the right column.

In experiment one, with the exception of the *NameAndPropertyAlignment* algorithm, all others generated higher precision in $M_{P2\text{-}T6}$. An average precision of 0.4462 was found in $M_{P2\text{-}T6}$, which is an improvement by 17.64% compared to M_B (at 0.3793). Improvement in recall can be seen in all matching algorithm in this trial, an average of 0.7501 was found in $M_{P2\text{-}T6}$ which is a 33.00% increase compared to M_B (at 0.5640). A similar finding is shown in the f-measure scores, whereby increased f-measure was found in $M_{P2\text{-}T6}$ by all matching algorithms. An average f-measure of 0.5062 was found $M_{P2\text{-}T6}$ which is an increase by 33.84% compared to M_B (at 0.3782). This improvement of the overall quality is further supported by the paired t-test carried out on the f-measure scores. With a p-value of 0.011, the t-test validates the statistical significance of the findings.

In experiment two, improvements of precision can be seen in all matching algorithms. An average precision of 0.7650 was found in M_{P2-T6}' which is a 10.58% increase compared to M_B' (at 0.6918). Increase recall can be seen with most matching algorithms with the exception of the *NameAndPropertyAlignment* algorithm. An average of 0.6675 was found in M_{P2-T6}' which is a 10.20% increase compared to M_B' (at 0.6057). Overall, increased f-measure is seen in all matching algorithms, where an average f-measure of 0.7037 was found in M_{P2-T6}' which is a 10.87% improvement compared to M_B' (at 0.6347). This improvement is supported by the paired t-test, with a p-value of 0.001, the null hypothesis is rejected.



Legend: 1 NameAndPropertyAlignment 5 SMOANameAlignment
2 StrucSubsDistAlignment 6 SubsDistNameAlignment
3 ClassStructAlignment 7 EditDistNameAlignment
4 NameEqAlignment 8 StringDistAlignment

Figure 5-20. Precision, Recall, F-Measure in Trial Six

The evaluation of the matches' confidence levels are presented in table 5-17. Scatter plots generated using this data can be found in appendix E, section E.6, figure E-6. In experiment one, more confident and less dispersed matches were found in M_{P2-T6} compared to M_B. An increased average confidence mean by 32.42% (at 0.9310) and a decreased average standard deviation by 5.44% (at 0.0940) were found in M_{P2-T6} compared to M_B. In experiment two, the matches in M_{P2-T6}' contained less dispersed confidence levels, however, are less confident on average compared to M_B'. An decrease average confidence mean by 0.12% (at 0.9470) as well as a decreased average standard deviation by 6.96% (at 0.1123) were found in M_{P2-T6}' compared to M_B'.

Table 5-17. Confidence Data from Trial Six

Exp.	Matching Technique	Baseline		SOCOM++ Trial 6 – Adjust Pseudo Feedback	
		St.Dev.	Mean	St.Dev.	Mean
i	1 NameAndPropertyAlignment	0.1014	0.9374	0.0943	0.9597
	2 StrucSubsDistAlignment	0.2505	0.7505	0.2381	0.7438
	3 ClassStructAlignment	0.2505	0.7505	0.0442	0.9785
	5 SMOANameAlignment	0.0582	0.9649	0.0442	0.9785
	6 SubsDistNameAlignment	0.1618	0.9041	0.1369	0.9272
	7 EditDistNameAlignment	0.0123	0.9909	0.0061	0.9984
	Avg.	**0.1391**	**0.8830**	**0.0940**	**0.9310**
ii	1 NameAndPropertyAlignment	0.0909	0.9674	0.1067	0.9628
	2 StrucSubsDistAlignment	0.1509	0.9059	0.1663	0.8998
	3 ClassStructAlignment	0.1545	0.9440	0.1099	0.9495
	5 SMOANameAlignment	0.1556	0.9431	0.1038	0.9557
	6 SubsDistNameAlignment	0.1541	0.9372	0.1700	0.9227
	7 EditDistNameAlignment	0.0179	0.9913	0.0170	0.9913
	Avg.	**0.1207**	**0.9481**	**0.1123**	**0.9470**

Table 5-18. Key Findings of Baseline, SOCOM++ Trial One and Six

Evaluations		Baseline	SOCOM++ Trial 1 (default configuration)	SOCOM++ Trial 6
Exp.1	Precision	0.3793	0. 4155	0.4462
	Recall	0.5640	0.6488	0.7501
	F-Measure	0.3782	0.4654	0.5062
	Confidence Level Mean	0.8830	0.9646	0.9310
	Confidence Level St.Dev.	0.1391	0.0613	0.0940
Exp.2	Precision	0.6918	0.7394	0.7650
	Recall	0.6057	0.6261	0.6675
	F-Measure	0.6347	0.6684	0.7037
	Confidence Level Mean	0.9481	0.9571	0.9470
	Confidence Level St.Dev.	0.1207	0.1065	0.1123

Table 5-18 presents the key findings of the baseline system, SOCOM++ trial one and trial six. In summary, trial six has improved the precision, recall and f-measure in both experiments compared to the SOCOM++ default configuration (i.e. trial one). However, the trade-offs on confidence levels are evident (i.e. increased standard deviation and decreased confidence level mean in trial six compared to trial one). This trade-off on confidence levels was shown previously in both trial four (optimising correctness) and trial five (optimising completeness). This consistent finding regarding the trade-off suggests that the feedback feature is able to improve the precision, recall and f-measure, however it may not be able to improve the confidence levels of the matches generated in the second iteration of SOCOM++.

5.4.4. Conclusions arising out of the Six Trials

This section presents a summary of the findings from the six trials carried out in the evaluation of SOCOM++. An overview of the findings in each trial from experiment one is presented in table 5-19. An overview of the findings in each trial from experiment two is presented in table 5-20.

In experiment one, trial three (where the AOLT selection process does not consider the semantic surroundings) was least successful at improving matching quality. The biggest improvement on precision (by 23.81% compared to the baseline system) was seen in trial five (optimising completeness). The highest recall (at 0.7501) was achieved in trial six (pseudo feedback with 0.5 cut-off point). Overall, the highest f-measure score (at 0.5098) was seen in trial five. The highest average confidence mean (at 0.9310) was achieved in trial six, and the lowest standard deviation (at 0.0613) was achieved in trial one (default SOCOM++ configuration). Increased f-measure can be seen in trial four (optimising correctness), however, the paired t-test on f-measure generated in M_B and $M_{P2\text{-}T4}$ fits the null hypothesis, and the p-value suggests that there is not enough evidence to conclude a difference between the baseline system and SOCOM++ trial four.

Table 5-19. An Overview of the SOCOM++ Trials in Experiment One

Best Results Achieved — Poorer Results in $M_{P2\text{-}T1/2/3/4/5/6}$ when compared to M_B — T-Test Result Fits Null Hypothesis

Exp. i	M_B (Avg.)					$M_{P2\text{-}T1/2/3/4/5/6}$ (Avg.)					% Change (+/-)					P-Value
	Precision	Recall	F-Measure	Mean	St.Dev.	Precision	Recall	F-Measure	Mean	St.Dev.	Precision	Recall	F-Measure	Mean	St.Dev.	
Trial 1	.3793	.5640	.3782	.8830	.1391	.4155	.6488	.4654	.9646	.0613	+9.54	+15.04	+23.06	+9.24	-55.93	.044
Trial 2						.4437	.6616	.4674	.9326	.1088	+16.98	+17.30	+23.59	+5.62	-21.78	.019
Trial 3						.3769	.4848	.3457	.8735	.1540	-0.63	-14.04	-8.59	-1.08	+10.71	.050
Trial 4						.4497	.6677	.4800	.9472	.0832	+18.56	+18.39	+26.92	+7.27	-40.19	.060
Trial 5						.4696	.7165	.5098	.9252	.0973	+23.81	+27.04	+34.80	+4.78	-30.05	.016
Trial 6						.4462	.7501	.5062	.9310	.0940	+17.64	+33.00	+33.84	+32.42	-5.44	.011

Table 5-20. An Overview of the SOCOM++ Trials in Experiment Two

Best Results Achieved — Poorer Results in $M_{P2\text{-}T1/2/3/4/5/6}$' when compared to M_B' — T-Test Result Fits Null Hypothesis

Exp. ii	M_B' (Avg.)					$M_{P2\text{-}T1/2/3/4/5/6}$' (Avg.)					% Change (+/-)					P-Value
	Precision	Recall	F-Measure	Mean	St.Dev.	Precision	Recall	F-Measure	Mean	St.Dev.	Precision	Recall	F-Measure	Mean	St.Dev.	
Trial 1	.6918	.6057	.6347	.9481	.1207	.7394	.6261	.6684	.9571	.1065	+6.88	+3.37	+5.31	+0.95	-11.76	.023
Trial 2						.7569	.6521	.6886	.9152	.1435	+9.41	+7.66	+8.49	-3.47	+18.89	.006
Trial 3						.7105	.6224	.6529	.9320	.1304	+2.70	+2.76	+2.87	-1.70	+8.04	.148
Trial 4						.7449	.6572	.6892	.9436	.1182	+7.68	+8.50	+8.59	-0.47	-2.07	.010
Trial 5						.7288	.6379	.6715	.9441	.1205	+5.35	+5.32	+5.80	-0.42	-0.17	.004
Trial 6						.7650	.6675	.7037	.9470	.1123	+10.58	+10.20	+10.87	-0.12	-6.96	.001

Table 5-21 shows a ranked list of all the CLOM systems in experiment one (including the baseline system, prototype one: SOCOM, and all six trial configurations of SOCOM++) with regards to all evaluation aspects (including precision, recall, f-measure, confidence level mean and standard deviation) during the mapping of the CSWRC ontology to the ISWC ontology. Depending on the ranking criteria, the systems are ranked in different orders as shown in table 5-21. Although SOCOM++ configuration one and configuration five are both ranked first twice (configuration one is ranked first due to its highest confidence level mean and lowest standard deviation; configuration five is ranked first by precision as well as by f-measure), it is difficult to declare a system as a clear winner since configuration one is only ranked sixth by precision, fifth by recall and f-measure, whereas configuration five is ranked fifth by confidence level mean and fourth by standard deviation. However, conclusions that can be drawn from the rankings include:

(1) the trials that had a second iteration of the AOLT process (i.e. SOCOM++ trial four, five and six discussed in section 5.4.3) generated higher precision, recall and f-measure compared to those that did not (i.e. the baseline system, SOCOM, SOCOM++ trial one, two and three discussed in section 5.4.2).

(2) when SOCOM++ trial three did not take account of the embedded semantics (i.e. semantic surroundings) during the AOLT selection process, it generated the worst results (ranked last in every evaluation aspect) in the CLOM experiments. This finding provides evidence that translations of ontology labels should not take place in isolation of the ontologies involved in a CLOM scenario.

(3) With the exception of trial three, all other configurations of SOCOM++ generated better results (across all evaluation criteria) than the baseline system. This finding further validates the effectiveness of the AOLT process, which is the core of this thesis.

Table 5-21. Rankings of CLOM Systems in Experiment One

Rank	Ranking Criteria				
	Precision	**Recall**	**F-Measure**	**Confidence Level Mean**	**Confidence Level St.Dev.**
1	SOCOM++ Trial 5 = 0.4696	SOCOM++ Trial 6 = 0.7501	SOCOM++ Trial 5 = 0.5098	SOCOM++ Trial 1 = 0.9646	SOCOM++ Trial 1 = 0.0613
2	SOCOM++ Trial 4 = 0.4497	SOCOM++ Trial 5 = 0.7165	SOCOM++ Trial 6 = 0.5062	SOCOM++ Trial 4 = 0.9472	SOCOM++ Trial 4 = 0.0832
3	SOCOM++ Trial 6 = 0.4462	SOCOM++ Trial 4 = 0.6677	SOCOM++ Trial 4 = 0.4800	SOCOM++ Trial 2 = 0.9326	SOCOM++ Trial 6 = 0.0940
4	SOCOM++ Trial 2 = 0.4437	SOCOM++ Trial 2 = 0.6616	SOCOM++ Trial 2 = 0.4674	SOCOM++ Trial 6 = 0.9310	SOCOM++ Trial 5 = 0.0973
5	SOCOM = 0.4367	SOCOM++ Trial 1 = 0.6488	SOCOM++ Trial 1 = 0.4654	SOCOM++ Trial 5 = 0.9252	SOCOM++ Trial 2 = 0.1088
6	SOCOM++ Trial 1 = 0.4155	SOCOM = 0.5854	SOCOM = 0.4146	SOCOM = 0.8962	SOCOM = 0.1239
7	Baseline = 0.3793	Baseline = 0.5640	Baseline = 0.3782	Baseline = 0.8830	Baseline = 0.1391
8	SOCOM++ Trial 3 = 0.3769	SOCOM++ Trial 3 = 0.4848	SOCOM++ Trial 3 = 0.3457	SOCOM++ Trial 3 = 0.8735	SOCOM++ Trial 3 = 0.1540

Table 5-22. Rankings of CLOM Systems in Experiment Two

Rank	Ranking Criteria				
	Precision	**Recall**	**F-Measure**	**Confidence Level Mean**	**Confidence Level St.Dev.**
1	SOCOM++ Trial 6 = 0.7650	SOCOM++ Trial 6 = 0.6675	SOCOM++ Trial 6 = 0.7037	SOCOM = 0.9640	SOCOM++ Trial 1 = 0.1065
2	SOCOM++ Trial 2 = 0.7569	SOCOM++ Trial 4 = 0.6572	SOCOM++ Trial 4 = 0.6892	SOCOM++ Trial 1 = 0.9571	SOCOM = 0.1110
3	SOCOM++ Trial 4 = 0.7449	SOCOM++ Trial 2 = 0.6521	SOCOM++ Trial 2 = 0.6886	Baseline = 0.9481	SOCOM++ Trial 6 = 0.1123
4	SOCOM++ Trial 1 = 0.7394	SOCOM++ Trial 5 = 0.6379	SOCOM++ Trial 5 = 0.6715	SOCOM++ Trial 6 = 0.9470	SOCOM++ Trial 4 = 0.1182
5	SOCOM++ Trial 5 = 0.7288	SOCOM = 0.6353	SOCOM++ Trial 1 = 0.6684	SOCOM++ Trial 5 = 0.9441	SOCOM++ Trial 5 = 0.1205
6	SOCOM++ Trial 3 = 0.7105	SOCOM++ Trial 1 = 0.6261	SOCOM = 0.6621	SOCOM++ Trial 4 = 0.9436	Baseline = 0.1207
7	SOCOM = 0.7084	SOCOM++ Trial 3 = 0.6224	SOCOM++ Trial 3 = 0.6529	SOCOM++ Trial 3 = 0.9320	SOCOM++ Trial 3 = 0.1304
8	Baseline = 0.6918	Baseline = 0.6057	Baseline = 0.6347	SOCOM++ Trial 2 = 0.9152	SOCOM++ Trial 2 = 0.1435

In experiment two, all six configurations of SOCOM++ generated higher precision, recall and f-measure compared to the baseline system, although improvement on confidence levels (i.e. increased confidence level mean, decreased standard deviation) is not always evident. Table 5-22 presents the ranked list of all CLOM systems when mapping the 101 ontology to the 206 ontology. In terms of precision, recall and f-measure, it is clear that trial six generated the best results (ranked first) whereas the baseline system generated the worst results (ranked last). In terms of confidence level mean and standard deviation, it is however difficult to identify a clear winner. Nevertheless, the results from this experiment clearly demonstrate that CLOM systems that incorporate the AOLT process (i.e. SOCOM, and six configurations of SOCOM++) generated matches with higher precision, recall and f-measure than the baseline system. This finding further supports the AOLT concept proposed in this thesis.

The experiments shown in the six SOCOM++ trials are somewhat limited in their domains and natural language pairs covered. However, as examples of CLOM scenarios that involve ontologies with distinct and similar characteristics, the findings from these experiments are nonetheless useful to gain an insight into the AOLT process. Table 5-23 shows the ranks achieved by all CLOM systems in both experiments, which summarises the ranks presented in table 5-21 and table 5-22. For example, SOCOM++ trial one configuration achieved rank one 3 times; rank two 1 time; rank four 1 time, rank five 3 times and rank six 2 times. Assuming precision, recall, f-measure, confidence level mean and standard deviation are as important as one another, the average rank that is achieved by each CLOM system can be calculated: for SOCOM++ trial one, its average rank is $(1\times3 + 2\times1 + 4\times1 + 5\times3 + 6\times2)/(3 + 1 + 1 + 3 + 2) = 3.6$. In table 5-23, the highest average rank is achieved by the SOCOM++ trial six configuration (with an average rank of 2.3), and the lowest average rank is achieved by the SOCOM++ trial three configuration (with an average rank of 7.4). In order of best to worst average rank achieved, the CLOM systems can be ordered as SOCOM++ trial six in first place, followed by the SOCOM++ trial four in second place, then the SOCOM++ trial five, the SOCOM++ trial one, the SOCOM++ trial two, the SOCOM system, the baseline system, and the SOCOM++ trial three in last place. Note though SOCOM++ trial one and five both achieved an average rank of 3.6, the trial five configuration is considered better as it contains a better rank record (i.e. it has a better record with ranks in fifth place or higher) compared to trial one (i.e. with a poorer record with ranks in sixth place twice).

Table 5-23. An Overview of Ranks achieved in Experiment One and Two

System \ Rank	1	2	3	4	5	6	7	8	Avg. Rank
SOCOM++ Trial 1	×3	×1	-	×1	×3	×2	-	-	3.6
SOCOM++ Trial 2	-	×1	×3	×3	×1	-	-	×2	4.4
SOCOM++ Trial 3	-	-	-	-	-	×1	×4	×5	7.4
SOCOM++ Trial 4	-	×5	×3	×1	-	×1	-	-	2.9
SOCOM++ Trial 5	×2	×1	-	×3	×4	-	-	-	3.6
SOCOM++ Trial 6	×4	×1	×3	×2	-	-	-	-	2.3
SOCOM	×1	×1	-	-	×2	×5	×1	-	5.0
Baseline	-	-	×1	-	-	×1	×5	×3	6.8

Three key conclusions can be drawn from the analysis on the rankings achieved. First of all, SOCOM++ trial six can be considered as the overall best configuration in the experiments conducted. Compared to the other two trials (SOCOM++ trial four and five) that also carried out a second iteration of the AOLT process, trial six is a better way to assess matches generated in the first iteration. While trial four applies a strict cut-off point (assuming matches with 1.0 confidence levels are correct) and trial five applies no cut-off point (assuming all matches are correct), trial six is relaxed yet effective by applying the 0.5 cut-off point (since 0.5 is a natural division point between 0.0 and 1.0). This assumption is more useful to determine the AOLT results in the second iteration as it concentrates on the incline in the confidence levels (i.e. equal or greater than 0.5 shows an incline towards confident while less than 0.5 shows an incline towards not confident) rather than treating the confidence levels as precise assessments on the matches' correctness (as seen with trial four and five).

Secondly, the CLOM systems that carried out a second iteration of the AOLT process (i.e. SOCOM++ trial four, five and six) achieved better rankings than those that did not (SOCOM++ trial one, two, three, the SOCOM and the baseline system). This finding suggests that using a form of feedback for the AOLT process (whether by optimising correctness in trial four, or optimising completeness in trial five or applying pseudo feedback in trial six) can further improve the AOLT results even more which consequently leads to better mapping quality.

Lastly, it is shown that SOCOM++ trial three achieved the worst rankings in the experiments. This finding is in fact further evidence to support the AOLT concept (i.e. translations should not take place in isolation of the mapping context), since trial three did not consider the mapping context (i.e. the semantic surroundings) during the AOLT process, even though the candidate translation pool was increased (i.e. synonyms were available for both candidate translations of the source labels and target labels), the translations were still poor which led to low mapping quality.

In conclusion, configurable inputs of the AOLT process have been successfully demonstrated in the six trials. It is shown through the evaluations that the AOLT process in SOCOM++ can be adjusted in order to alter its output which will lead to a variety of mapping outcomes. What is not yet known is how SOCOM++ might cope with increased workload such as larger ontologies. This is investigated next.

5.4.5. Scalability Tests

The trials shown thus far successfully demonstrate an improved prototype: SOCOM++ from the initial prototype: SOCOM, in terms of the mapping quality achieved (i.e. precision, recall, f-measure, confidence level mean and standard deviation) in the same CLOM experiments (i.e. mapping the CSWRC ontology in Chinese to the ISWC ontology in English, and mapping the 101 ontology in English to the 206 ontology in French). However, these experimental findings cannot identify major workloads that may be potentially improved for future prototypes of the proposed system. To address this shortcoming, scalability tests are carried out which aim to identify major workloads in SOCOM++ that can be improved to mitigate bottlenecks in future prototypes. In particular, this section investigates the execution time required by SOCOM++ when working with increased workload (i.e. larger ontologies, sophisticated configurations of the AOLT selection process). Section 5.4.5.1 discusses the experimental setup of the scalability tests. The findings are presented in section 5.4.5.2.

5.4.5.1. Tests Setup

The goal of the scalability tests is to investigate how examples of a simple (e.g. configuration used in SOCOM++ trial two) and a sophisticated (e.g. configuration used in SOCOM++ trial four) configuration of SOCOM++ will cope with smaller and larger ontologies in terms of execution time required. The ontologies used in the scalability tests are discussed next.

The CSWRC ontology in Chinese (a total of 128 entities, see chapter 4, section 4.5.1.1) and the ISWC ontology (a total of 118 entities, see chapter 3, section 3.4) in English of the research domain are used as an example of small ontologies in the scalability tests. To represent a larger ontology pair, an OWL ontology in English and an OWL ontology in Japanese of the automobile domain are taken from the OAEI 2008 multilingual directory data set[99]. The English ontology was constructed using the Google Directory[100], the Open Directory Project[101] and the Yahoo Directory[102]. It contains 867 classes, 4089 individuals and no properties. The Japanese ontology is constructed using the Lycos Japan[103] and the Yahoo Japan Directory[104]. It contains 1063 classes, 2727 individuals and no properties. For more information on how these ontologies are generated, see [Ichise et al., 2003] and [Ichise et al., 2004]. It can be argued that since these ontologies do not contain properties, they are rather classifications than ontologies as such. Nevertheless, this author views them as OWL vocabularies with less sophisticated/expressive restrictions. As the goal of the scalability tests is to investigate the execution time required by SOCOM++ when working with increased ontology sizes, these automobile ontologies are suitable for this purpose considering the larger set of classes and individuals (compared to the CSWRC and the ISWC ontology) that will need to be processed by SOCOM++. Table 5-24 gives an overview of the characteristics of the two pairs of ontologies in the scalability tests. In terms of total entity count, the automobile ontologies contain over 35 times more entities than the research ontologies.

Table 5-24. Ontologies used in the Scalability Tests

Test	**Ontology**	**Natural Language**	**Ontology Size**				
			Class	Data Type Property	Object Property	Individual	Total
i	O_1	Chinese	54	30	44	0	128
	O_2	English	33	17	18	50	118
ii	O_1	Japanese	1063	0	0	2727	3790
	O_2	English	867	0	0	4089	4956

To match the ontologies above, SOCOM++ is executed using two different configurations (see section 5.4.1 table 5-2): the settings used in trial two (a single iteration of the system where synonyms are not considered during the AOLT selection process) and the settings used in trial five (two iterations of the system where the feedback feature does not enforce any cut-off point, in addition, the AOLT process will considered all available inputs). In trial two, the AOLT selection process only needs to process a minimum set of candidate AOLT results (see section 5.4.2.2). In trial five, the AOLT selection process not only needs to process a maximum set of candidate AOLT results but also needs to carry out two iterations of the AOLT process (for which the second iteration considers all matches from the first iteration as correct, see section

[99] http://oaei.ontologymatching.org/2008/mldirectory/
[100] http://www.google.com/dirhp?hl=en
[101] http://www.dmoz.org/
[102] http://dir.yahoo.com/
[103] http://www.lycos.co.jp/
[104] http://dir.yahoo.co.jp/

5.4.3.2. This is the most demanding type of iteration compared to trial four or six while only partial matches are considered correct). These trial configurations are selected in the scalability tests to represent a simpler configuration (trial two) and a more sophisticated configuration (trial five) of SOCOM++.

In the scalability tests carried out, SOCOM++ ran on Windows Vista Business edition, service pack 2 that was installed on a Dell Latitude D830 notebook powered by Intel® Core™ 2 Duo CPU T7500 @ 2.20 GHz with 2.00 GB memory. The runtime environment was provided by the Eclipse[105] Europa version 3.3.2 platform which ran on JRE6 (Java Runtime Environment 6) JVM (Java Virtual Machine). The time taken to complete each stage (i.e. the semantic analysis, the ontology rendition and the ontology mapping step discussed in section 5.3) in the CLOM process and the total execution time is recorded and considered as a measurement for efficiency for the two trial configurations. To measure the execution time, `System.currentTimeMillis()` is added in the program code at the start (the current time in milliseconds when the application is initiated) and the end (the current time in milliseconds when the application is terminated) of each stage of the CLOM process, whereby the difference between the end and the start is the time took (in milliseconds) to complete a stage. The execution time is finally converted in minutes (from milliseconds).

5.4.5.2. Findings and Analysis

This section presents the findings and conclusions drawn from the scalability tests. In a given mapping scenario, the two configurations (trial two and trial five) are compared to each other first. Then for the same trial configuration, comparisons are made between the different mapping scenarios.

As shown in table 5-25, when mapping the Chinese CSWRC ontology to the English ISWC ontology (test i), it took the simpler configuration (i.e. trial two) 1.4899 minutes to complete semantic analysis (which includes generating and storing candidate translations for the O_1 labels, the semantic surroundings for all entities presented in both ontologies) and 4.8498 minutes to complete ontology rendition (which includes selecting and storing AOLT results for each O_1 label, and rendering O_1'). Depending on the MOM algorithm used, there is a slight variation in the ontology mapping step (which includes generating and storing matches between O_1' and O_2, and converting

[105] http://www.eclipse.org/

these matches to CLOM results between O_1 and O_2 based on the selected AOLT results). However, the distinctions among them are very little: for instance, the quickest time took to complete the ontology mapping stage was 0.0108 minutes (when applying the *StringDistAlignment* algorithm), and the slowest time took was 0.0496 minutes - this is only a difference by 2.3280 seconds (0.0388 minutes).

Table 5-25. Scalability Tests' Results

Matching Algorithms:

1 NameAndPropertyAlignment
2 StrucSubsDistAlignment
3 ClassStructAlignment
4 NameEqAlignment
5 SMOANameAlignment
6 SubsDistNameAlignment
7 EditDistNameAlignment
8 StringDistAlignment

Test			**i. Map the CSWRC Ontology (Chinese) to the ISWC Ontology (English)**		**ii. Map Automobile Ontologies (Japanese to English)**	
SOCOM++ Configuration			Trial 2	Trial 5	Trial 2	Trial 5
Execution Time (Mins)	Semantic Analysis		1.4899	5.6881	230.8115	1258.7525
	Ontology Rendition	1	4.8498	5.7146	210.2826	540.9799
		2		5.5919		540.7425
		3		5.6415		542.3158
		4		5.6442		541.8760
		5		5.5882		540.5479
		6		5.5796		541.5472
		7		5.7536		542.8751
		8		5.6569		541.8765
		Avg.	n/a	5.6463	n/a	541.5951
	Ontology Mapping	1	0.0496	0.0130	0.2636	0.2693
		2	0.0248	0.0122	0.2592	0.2397
		3	0.0120	0.0132	0.9129	1.2522
		4	0.0125	0.0113	0.2198	0.3940
		5	0.0148	0.0148	0.6554	0.9143
		6	0.0117	0.0123	0.3935	0.5653
		7	0.0130	0.0128	27.5276	6.0223
		8	0.0108	0.0114	0.5773	0.4255
		Avg.	0.0187	0.0126	3.8512	1.2603
	Total (Avg.)		6.3584	11.3470	444.9453	1801.6079

The average total execution time is calculated as: *total execution time (Avg.) = time took to complete semantic analysis + time took to complete ontology rendition + average time took to complete ontology mapping* where a total average execution time of 6.3584 minutes was found for trial two in test scenario one. When applying the trial five configuration (synonyms are now generated for labels in both O_1 and O_2), increased execution times were found in the semantic analysis stage as thesauri are now included. As shown in table 5-25, it took over 5 minutes for trial five to complete semantic analysis. This is 3.8 times the execution time taken to complete semantic analysis in trial two. The ontology rendition stage also took longer to complete in trial five, as this configuration now requires more processing power when selecting from a much larger candidate AOLT pool. An average execution time of 5.6463 minutes was recorded for ontology rendition in trial five. Compared to trial two, this is an average increase by over 16%. An average time of 0.0126 minutes was achieved for ontology mapping in trial five, which is almost the same (the difference is only 36 seconds) with trial two. This is not a surprising finding, as the ontology size remains unchanged in trial two and trial five. An average total processing time of 11.3470 minutes was found in trial five, which is an increase by over 78% compared to trial two. This increase is largely due to the need to generate synonyms for the labels extracted from both O_1 and O_2 (at the semantic analysis stage). In addition, having to select AOLT results from a much bigger candidate pool (at the ontology rendition stage) also contributed towards the increase of the execution time.

When dealing with the much larger ontology pair (test ii) - mapping the Japanese automobile ontology to the English automobile ontology using the trial two configuration of SOCOM++, it took nearly four hours (at 230.8115 minutes) to complete the semantic analysis stage, a further three hours (at 540.9799 minutes) to complete the ontology rendition stage and about four minutes (at 3.8512 minutes on average) to complete the ontology mapping stage. Similar to the findings from test i, time took to generate the final mapping results vary depending on the MOM algorithm applied. However, the variations are small with the exception of the *EditDistName-Alignment* algorithm. This algorithm calculates the edit distances between all resources by creating a matrix of distances first and then generates matches based on these distances. This requirement on pre-processing the distances between all pairs of resources in the given ontologies is more time consuming compared to other algorithms.

In total, the trial two configuration took more than seven hours (at 444.9453 minutes) to complete the CLOM process in test scenario two. When the trial five configuration was applied, increased processing time was found especially in the semantic analysis and the ontology rendition stage. Over twenty hours (at 1258.7525 minutes) were required to complete the semantic analysis in trial five. This is over five times the time recorded for trial two. On average, approximately nine hours (at 541.5951 minutes) were required to complete the ontology rendition stage (as the assessment is conducted on a per-matching algorithm basis, time took to generate O_1' varies depending on the matching algorithm used). This is over double the time recorded in trial two. The ontology mapping stage remained relatively fast which took an average of just over one minute (at 1.2603 minute) to complete. In total, an average of 1801.6079 minutes (approximately thirty hours) were recorded in trial five, which is over four times of what was recorded in trial two.

When applying the trial two configuration, test two (larger ontologies) took almost $\times 70$ the total time needed in test one (smaller ontologies). More specifically, the semantic analysis stage took almost $\times 155$ the time; the ontology rendition stage took over $\times 43$ the time; and the ontology mapping stage took over $\times 205$ the time. It is clear that the increased total execution time is largely caused by the increased workload at all stages. This is further supported by the findings of trial five. When the trial five configuration was applied, test two took over $\times 158$ the time required in test one. Specifically, the semantic analysis stage took over $\times 221$ the time needed in test one; the ontology rendition stage took over $\times 95$ the time required in test one; and the ontology mapping stage took about 100 times the time required in test one.

In summary, increased processing times are shown when the system has to deal with large ontologies and especially when using sophisticated configurations of SOCOM++. To prevent out-of-memory error, heap memory size had to be increased in the second test for both simple (i.e. trial two) and sophisticated (i.e. trial five) configurations. The bottleneck of the system is at the semantic analysis and the ontology rendition step. The semantic analysis step needs to parse an entire ontology to extract entity labels in order to generate and store information such as candidate translations, synonyms and semantic surroundings; and the ontology rendition stage needs to process an increased candidate AOLT pool to overcome collisions and finally select the appropriate translations. The current implementation uses the Jena framework to parse ontologies, which showed significantly slower processing time when a reasoner is presented[106]. This may be improved with other parser tools (e.g. the alignment API integrates the OWL API[107] in the process of generating MOM matches, which was shown to be reasonably fast in the scalability test), however, further tests are necessary. The results from the scalability tests suggest that the current implementation is not suitable for CLOM scenarios that need to be conducted in real time. However, the current integration would be suitable if the execution times are not a critical issue.

Though it can be argued that other aspects such as the operating system, the hardware on which it is running can be changed to optimise the allocation of resources when running SOCOM++ besides tuning the application code and APIs, however, the scalability tests shown in this section are not an exhaustive list of performance tests, but

[106] Further reading regarding slow processing time when a reasoner is presented in the Jena framework can be found at: http://tech.groups.yahoo.com/group/jena-dev/message/42621
[107] http://owlapi.sourceforge.net/

rather examples of extreme AOLT configurations (i.e. one extreme being the simple configuration used in trial two and the other being the sophisticated configuration used in trial five) and how they cope with increased workload (i.e. from smaller ontologies such as the Chinese CSWRC and the English ISWC ontology, to the larger ontologies such as the automobile ontology in Japanese and English). The tests shown in this section are useful to identify major workloads and potential bottlenecks.

5.5. Summary

This chapter presents the design, implementation and evaluation of the second prototype: SOCOM++. SOCOM++ allows the user to adjust various inputs that are available to the AOLT process in an effort to influence the final CLOM outcome. SOCOM++ is demonstrated and evaluated in six trial configurations, where each trial focuses on one particular configurable feature. The first three trials concern the adjustment of ontology-related semantics, such as making use of all internal and background semantics (as demonstrated in trial one), generating AOLT results without any background semantics (i.e. synonyms, as demonstrated in trial two) and ignoring internal semantics (i.e. semantic surroundings, as demonstrated in trial three). The remaining three trials focus on executing two iterations of the AOLT process, whereby matches generated from the first iteration are assessed to assist with the selection of the AOLT results in the second iteration. Three approaches to achieve the assessment were demonstrated, including optimising correctness (as demonstrated in trial four), optimising completeness (as demonstrated in trial five) and pseudo feedback (as demonstrated in trial six). It is shown through these trials that various precision, recall and f-measure scores as well as confidence mean and standard deviation can be achieved with the same pair of ontologies in a CLOM setting through variations of the SOCOM++ configuration. This shows that depending on what is desired by the user, SOCOM++ can be configured accordingly to adapt to the particular CLOM scenario. To investigate the scalability of SOCOM++, two configurations of SOCOM++: trial two and five are used as examples of a lightweight and a heavyweight configuration in CLOM scenarios involving a smaller and a larger ontology pair. Execution times recorded in the scalability test highlight the major workloads in the system where potentially up to $\times 158$ the execution time is required for a sophisticated configuration than a simpler configuration. This may be improved by integrating other APIs in future implementations given that the operating system and the hardware remain unchanged.

6 CONCLUSIONS

6.1. Chapter Overview

This chapter discusses how well the objectives that are set out for this research have been achieved in section 6.2. The main contributions of this thesis are summarised in section 6.3, followed by suggestions for future research in section 6.4. Finally, section 6.5 concludes this thesis with some final remarks.

6.2. Objectives

To answer the research question identified in chapter 1 (section 1.3): *this research investigates the extent to which machine translation and monolingual ontology matching techniques can be incorporated to support the generation of quality mapping results in the process of cross-lingual ontology mapping*, this thesis has built upon the baseline approach to cross-lingual ontology mapping and proposed the AOLT concept to achieve ontology label translations specifically suited for the purpose of cross-lingual ontology mapping. Two AOLT-based, cross-lingual ontology mapping systems: SOCOM and SOCOM++ have been designed and developed in order to support the evaluation of the AOLT concept and the AOLT-based CLOM systems. It has been shown through the evaluation of SOCOM and SOCOM++ that the AOLT process is more effective (in terms of improved precision, recall, f-measure, confidence level mean and standard deviation) at improving cross-lingual ontology mapping quality than the baseline system. This thesis has also met the research objectives (discussed in chapter 1, section 1.4) set out for this research, as discussed next.

Research objective (1): conduct reviews on the state of the art in CLOM, MT, MOM and current approaches to the evaluation of mapping results.

This objective is achieved through surveying current approaches used in CLOM. Five categories of techniques including manual CLOM, corpus-based CLOM, CLOM via linguistic enrichment, CLOM via indirect alignment and translation-based CLOM were identified through this survey. This categorisation of CLOM approaches is the first attempt at classifying various current approaches to CLOM, and contributes to the literature by providing a consolidated view on existing CLOM strategies. To the best of this author's knowledge, the CLOM approaches included in the categorisation are a complete list of existing strategies to achieve cross-lingual ontology mapping at the time of this writing. Limitations of the approaches in each category are discussed (see chapter 2, section 2.4.1) and the translation-based approach to CLOM is established through the state of the art review as the most advanced technique to CLOM currently available. The translation-based approach applies translation techniques to turn a cross-lingual mapping problem into a monolingual mapping problem first, which can then be solved using MOM tools. Background reviews on MT and MOM have thus been carried out and state of the art tools in MT (e.g. the GoogleTranslate API, Microsoft translator API) and MOM (e.g. the Alignment API) that are suitable to facilitate the CLOM process were identified (see chapter 2, section 2.5 and 2.6). Finally, a survey on state of the art evaluation approaches in ontology mapping was conducted (see chapter 2, section 2.7). Through this review, precision, recall and f-measure were identified as the most practised metrics in mapping evaluation. In addition, paired t-test is identified as a suitable hypothesis testing technique when validating the difference between two systems. Furthermore, mean and standard deviation have been identified as appropriate tools to evaluate the confidence levels of the matches generated.

Research objective (2): design and develop a process specifically suited for translations carried out for the purpose of CLOM and implement a set of tools to support this translation process in order to achieve CLOM results via MOM techniques.

To achieve this second objective, it was important to first understand the limitations of the current translation-based approach to CLOM. Thus, a baseline system (i.e. a realisation of the translation-based approach to CLOM) was developed (see chapter 3, section 3.3) using the MT and MOM tools identified in the state of the art. Though the baseline implementation uses a limited number of MT and MOM techniques, however, is representative of translation-based approach to CLOM in the state of the art. This baseline CLOM system is sufficient for the purposes of this research, as it provides this thesis with a reference point for further development based on the current state of the art in CLOM. The baseline system was evaluated in two experiments that focus on the ontology label translations. These experiments are not designed to be an exhaustive list of all possible natural languages in ontologies, however, they are sufficient for the purposes of this thesis as they offer this author with an opportunity to gain an more in-depth understanding of the translation-based CLOM approach. Findings from these experiments show that the requirement for translations in the context of CLOM differs from those that take place in the context of localisation. In CLOM, translation noise is introduced when an incorrect match is generated or a correct match is neglected. This noise consequently leads to poor matching quality.

Motivated by addressing this shortcoming of the baseline system, the AOLT concept was proposed (discussed in chapter 4, section 4.2), which aims to select translations that are most likely to maximise the matching ability of the subsequent MOM step. To realise the proposed AOLT process, two prototypes of the Semantic-Oriented Cross-lingual Ontology Mapping system: SOCOM (discussed in chapter 4) and SOCOM++ (discussed in chapter 5) were designed and developed to facilitate the selection of the AOLT results. The key to the AOLT process is that ontology label translations are not taken place in isolation of the ontologies involved in a CLOM scenario. The embedded semantics (e.g. labels used in the target ontology, semantic surrounding of the entities in both source and target ontology) of the ontologies as well as background semantics (e.g. synonyms of the labels in the target ontology, synonyms of candidate translations for the labels in the source ontology) are used in the selection process in order to achieve appropriate translations. This AOLT concept is a novel approach to ontology label translations conducted in the context of CLOM.

To evaluate the AOLT process, an initial proof-of-concept CLOM system: SOCOM, that integrates a basic AOLT component was applied to two CLOM experiments. The basic AOLT process (in SOCOM) makes use of a minimum amount of semantics that are always available in a given CLOM scenario (i.e. the labels in the given ontologies and their semantic surroundings). The effectiveness of the basic AOLT process was shown through the evaluation of SOCOM, whereby improvement of the matching quality is seen with SOCOM compared to the baseline system.

Motivated by this positive finding, an improved second prototype: SOCOM++ was developed to gain further improvement on the matching quality given the same ontology pair. The goal of the AOLT process (in SOCOM++) is to influence the matching outcome by adjusting inputs to the AOLT process. The effectiveness of this more sophisticated AOLT process was evaluated through the CLOM results generated by SOCOM++. Six trials each with a different AOLT configuration were carried out. It was shown through these trials that the adjustment of the AOLT input is effective at altering CLOM outcomes. The scalability of the AOLT process (in terms of execution times) in SOCOM++ was also investigated (see chapter 5, section 5.4.5). It is shown that when workload doubles (e.g. increased ontology size and more complex configuration for the AOLT process), execution times of SOCOM++ also increases by at least double (depending on the specific AOLT configuration). As discussed in chapter 4 (section 4.2), there can be other ways to realise the AOLT concept such as expert-based or rule-based approaches. The AOLT process presented in SOCOM and SOCOM++ are not an exhaustive list of implementation options, but rather example realisations of the AOLT concept. In author's opinion, these example implementations are sufficient for the purpose of demonstrating and evaluating the proposed AOLT concept in this thesis. In summary, the second research objective has been achieved through the development of the AOLT-based CLOM process and the SOCOM, SOCOM++ tools which implement that process.

Research objective (3): evaluate the quality of the mappings generated using the set of tools in CLOM scenarios and demonstrate the use of the set of tools in a real-world application.

The evaluations carried out in this thesis use metrics (identified in chapter 2) that are currently being used in the state of the art in ontology mapping evaluation. In particular, precision, recall and f-measure which originated in the field of IR were applied. As the matches generated in this thesis are accompanied by confidence levels, they are also evaluated using mean and standard deviation. To evaluate SOCOM, two CLOM experiments were designed (see chapter 4, section 4.5). One experiment involves an ontology pair with natural languages from different language families, containing overlapping domains and different structures (i.e. the CSWRC ontology in Chinese and the ISWC ontology in English of the research domain). The other experiment involves another ontology pair with natural languages from the same language family, containing almost identical structures and domains (i.e. the OAEI 101 ontology in English and the OAEI 206 ontology in French of the bibliography domain). The baseline system and SOCOM were applied to these experiments and their CLOM results were compared to gold standards using the aforementioned metrics.

To demonstrate the potential application of SOCOM, a case study was conducted which enabled cross-lingual document retrieval of the ARCHING [Steichen et al., 2011] system via CLOM results (see chapter 4, section 4.6). The ontologies in this case study were generated using real-world data from Symantec's Norton 360 security product, one in English and the other in German with overlapping domains. CLIR is enabled by generating CLOM results between these ontologies. SOCOM's applicability in the real world was shown through the feasibility of this case study.

To evaluate SOCOM++, a total of six trials were carried out (see chapter 5, section 5.4). Each trial had a different configuration which focused on one particular AOLT input (see chapter 5, table 5-2), and was applied to the same two CLOM experiments (used in the evaluation of SOCOM). The evaluation results of these trials show an array of matching quality (e.g. even better than what was shown in SOCOM) that can be achieved depending on the adjustment of the AOLT process. To evaluate the scalability of this configurable AOLT process, two configurations of SOCOM++ (one with less sophisticated configuration, i.e. minimum AOLT input; and the other with more sophisticated configuration, i.e. maximum AOLT input and a second iteration) were applied to a smaller ontology pair with just over 200 entities, and a larger ontology pair with over 8000 entities collectively (discussed in chapter 5, section 5.4.5). It is shown through the scalability tests that the execution time of the system increases as the workload increases. The experiments used in the evaluation of SOCOM and SOCOM++ are not designed to be an exhaustive list of CLOM scenarios, but rather example settings designed to evaluate the effectiveness of the AOLT process. For the purposes of this thesis, these experiments are suitable in this research.

Though the ontologies experimented with in this thesis are somewhat limited in terms of domain, size and expressiveness, they are however designed as examples in typical CLOM scenarios considering it would be rather difficult to experiment with an exhaustive list of ontologies in the given scope. In total, five natural languages including Chinese, English, French, German and Japanese have been experimented with in this thesis, which are believed to be a good representation of diverse natural languages. However, SOCOM, SOCOM++ and the underlying AOLT process are designed to work with any natural language pairs. The natural languages shown in this thesis are not an exhaustive list of all natural languages that can be involved in CLOM, but rather example scenarios. The findings from these evaluations nevertheless are motivating and support the proposed AOLT concept, which is the aim of this thesis.

6.3. Contributions

The major contribution of this thesis is the proposed AOLT concept which has been successfully demonstrated and evaluated. It is a novel approach to ontology label translations that are carried out for the purpose of CLOM. This AOLT concept is shown to be more effective at facilitating MT and MOM techniques in CLOM systems (e.g. SOCOM and SOCOM++) compared to the baseline system (i.e. the current state of the art approach to CLOM) in sets of CLOM experiments. To this author's best knowledge, the proposed AOLT concept is the first of its kind in the field of CLOM. As there has not been any study on the impact of translations on the process of cross-lingual ontology mapping, this thesis fills a current gap in the research literature.

A minor contribution of this thesis is the AOLT process in SOCOM and SOCOM++. Though there are other ways to realise the AOLT concept in CLOM such as expert-based or rule-based approaches (discussed in chapter 4, section 4.2), the AOLT process (integrated in SOCOM and SOCOM++) is the first attempt at achieving appropriate translations in the context of CLOM. These example AOLT processes provide a reference point for future implementations to realise the AOLT concept in the context of CLOM. In addition, the evaluations of the AOLT process shown in this thesis consist of repeatable experiments and replicable results, which are essential for measuring the effectiveness of future research carried out in this domain.

A total of five publications have derived from this research, including two full conference papers, one conference poster and two workshop papers, discussed next. With a lack of attention placed on CLOM, the pressing need to facilitate ontology mappings carried out in the multilingual environment is identified in the paper outlined below. In particular, the challenge concerning the translation of ontology labels in the context of CLOM is highlighted. This publication stresses the need to seek support for cross-lingual ontology mapping and introduces the SOCOM system.

> Fu B., Brennan R., O'Sullivan D., *Multilingual Ontology Mapping: Challenges and a Proposed Framework*. In Proceedings of the Symposium on Matching and Meaning - a symposium at the AISB 2009 Convention, SSAISB: The Society for the Study of Artificial Intelligence and the Simulation of Behaviour, ISBN 1902956842, pp. 32-35, Edinburgh, UK, April 2009

Following this initial proposal of the AOLT concept, to demonstrate why the AOLT concept will be useful in CLOM and to provide evidence of the shortcomings of the current translation-based approach to CLOM, the paper below documents the design of the baseline system, its implementation and evaluation (discussed in chapter 3). This publication details the design, implementation and the evaluation of the baseline system and focuses on the impact of ontology label translations on the mapping quality. This paper evidently shows why the current translation-based approach to CLOM needs to be improved. This publication has influenced the implementation of the API described in [Trojahn et al., 2010], which shows the impact of this research on the on-going research effort in the field of cross-lingual ontology mapping.

> Fu B., Brennan R., O'Sullivan D., *Cross-lingual Ontology Mapping - An Investigation of the Impact of Machine Translation*. In Proceedings of the 4th Annual Asian Semantic Web Conference (ASWC 2009), LNCS 5926, pp. 1-15, Shanghai, China, December 2009

Motivated by improving CLOM quality and realising the proposed AOLT concept, the paper below documents the design of SOCOM that integrates the basic AOLT process, its implementation and evaluation (discussed in chapter 4). The preliminary findings shown in this paper have successfully demonstrated the potential of the basic AOLT process at improving CLOM quality. In addition, this publication discusses how CLOM can be beneficial for systems and applications on the semantic web. More specifically, this paper proposes a novel approach to CLIR that is enabled through the use of CLOM results. The importance of tackling multilinguality on the semantic web is clearly recognised by the research community given the success of the first Multilingual Semantic Web workshop that was collocated at the World Wide Web conference in 2010. The paper below is closely connected with the theme of the workshop and contributes to the advancements in techniques aimed for the multilingual semantic web.

> Fu B., Brennan R., O'Sullivan D., *Cross-Lingual Ontology Mapping and Its Use on the Multilingual Semantic Web*. In Proceedings of the 1st Workshop on the Multilingual Semantic Web, collocated at the 19th International World Wide Web Conference (WWW 2010), CEUR Vol. 571, pp. 13-20, Raleigh, USA, April 2010

Centred on the specific strategy that can be used to evaluate the SOCOM system, the paper below introduces a set of two CLOM experiments (i.e. mapping the Chinese CSWRC ontology to the English ISWC ontology, and mapping the English OAEI 101 ontology to the French OAEI 206 ontology, discussed in chapter 4, section 4.5). This publication focuses on the evaluation approach undertaken for SOCOM, and adds confidence to the follow-up experiments carried out in this thesis.

> Fu B., Brennan R., O'Sullivan D., *Evaluation of a Semantic-Oriented Approach to Cross-Lingual Ontology Mapping*. In Proceedings of the EKAW 2010 Poster and Demo Track (EKAW 2010 - Knowledge Engineering and Knowledge Management by the Masses), CEUR-WS Vol.674, Lisbon, Portugal, October 2010

Built upon the success of SOCOM, an improved prototype SOCOM++ was designed and implemented that integrates a more sophisticated AOLT process (discussed in chapter 5). Among various adjustable inputs to the AOLT process, the following paper focuses on the specific pseudo feedback feature of the SOCOM++ system, and presents the evaluation of this feature using the OAEI dataset (i.e. mapping the English OAEI 101 ontology to the French OAEI 206 ontology). This publication successfully demonstrates how mapping quality can be improved even more (compared to the initial SOCOM system) given two iterations of the AOLT process.

> Fu B., Brennan R., O'Sullivan D., *Using Pseudo Feedback to Improve Cross-Lingual Ontology Mapping*. In Proceedings of the 8th Extended Semantic Web Conference (ESWC 2011), LNCS 6643, pp. 336-351, Heraklion, Greece, May 2011

The adjustable nature of the SOCOM++ system - in other words, the inputs to the AOLT process can be configured to adjust the AOLT output which in turn alters the mapping outcome - that have been successfully demonstrated through the six trials discussed in chapter 5 (section 5.4) is documented in a journal paper submitted to the Journal of Web Semantics. This paper aims to contribute to the future development and advancement of CLOM systems by presenting the latest results derived from this research. Lastly, a notable contribution of the publications listed above is that the experiments and their results are repeatable and replicable, thus provides researchers conducting further research in this field with a clear reference point.

6.4. Future Work

The research shown in this thesis is the first attempt that focuses on improving CLOM quality through the use of appropriate translations. Building upon the results shown in this thesis, this research has also opened up several research opportunities for future work, discussed next.

Evaluation: firstly, though five natural languages (including Chinese, English, French, German and Japanese) have been experimented with in this thesis, this is however a small sample size. Additional evaluation experiments with more ontology pairs involving additional domains and natural languages will give further insight into the use of the AOLT process in CLOM. Secondly, more case studies should be developed and evaluated with task-oriented evaluation approaches such as Noy & Musen's user-centric evaluation strategy, or Hollink et al.'s end-to-end evaluation strategy as discussed in chapter 2 (section 2.7).

Implementation: firstly, the improvements are shown in a variety of matching techniques that are at the element-level as well as the structure-level. However, these matching techniques are from the same API. It is not yet known whether the same level of improvement (if there is an improvement) can be seen with other MT and MOM tools. Thus, further experiments are necessary. Secondly, the use of feedback in CLOM can be expanded to incorporate explicit and implicit feedback, whereby user knowledge (e.g. obtain the assessment of a match by explicitly asking the user) and user behaviours (e.g. obtain the assessment of a match by inferring from the user's previous assessment) may be used to assist the generation of reliable mappings. Thirdly, the pseudo feedback can be further extended to include negative feedback (e.g. a blacklist as opposed to the current whitelist shown in SOCOM++) whereby the system will recognise which MT tools should not be used in the second iteration. Fourthly, the impact of feedback (e.g. when the assumptions made on the matches are simply invalid) on the CLOM quality is not yet investigated in this thesis, future research could explore this area. For instance, given the feedback that was generated automatically by the system, evaluations on their soundness can be carried out by human experts which will identify the incorrect feedback. When a second iteration is executed using these incorrect feedback, the impact on the mapping quality (i.e. precision, recall, f-measure, confidence level mean and standard deviation) can be quantified. Fifthly, only two iterations of SOCOM++ is demonstrated in this thesis, further iterations using feedback can be evaluated in order to investigate whether a third, fourth etc. iteration can further improve mapping quality or not. Lastly, further development based on this research can expand to support graphical user interface in the process of facilitating mapping experts with CLOM tasks, as well as providing open-source API to help the advancement of this field.

Other approaches to CLOM: the current translation-based approach to CLOM is conditioned upon the AOLT results to generate desired mappings. This approach tailors the AOLT outcome to suit specified MOM techniques. In any CLOM scenario however, there will always be a finite set of candidate translations available. Though this could be a very large pool to choose from, nevertheless, it is still finite. In other words, as long as the CLOM process requires identifying the precise translations for ontology labels, i.e. the existence of O_1', the mapping outcome will be limited to a finite set of possible AOLT outcomes which in turn restricts the improvement that can be seen in a given CLOM scenario. Other approaches to CLOM that do not rely on O_1' or require the subsequent MOM step may be useful to explore in future research. Furthermore, future approaches could investigate the benefits of systems that use localised ontologies in the CLOM process, whereby conceptualisation mismatches have already been addressed by adapting the naming and the structure of ontological concepts to the target community.

Community: the advancement in the field of CLOM relies on the research community. Collaborate effort is required on several aspects. Firstly, CLOM data sets that are accompanied by readily available gold standards are limited, which makes the evaluation of CLOM techniques difficult. The Chinese CSWRC ontology that was generated during this research, as well as the gold standard generated between the CSWRC ontology and the ISWC ontology have been made available online. More contributions from others would help fostering innovations in the field of CLOM. Secondly, there is a lack of workshops and contests organised specifically for this field. The progress of these aspects requires the dedication from the research community as a whole. This work presents some motivating findings and issues, and it is the author's intent to continue contributing to the field of CLOM.

6.5. Final Remarks

Addressing multilinguality is recognised as one of the pressing challenges for the semantic web [Benjamins, 2004]. It is the author's opinion that cross-lingual ontology mapping can provide a solution to multilingual semantic interoperability, and assists with dealing with multilinguality on the semantic web. To improve the quality of CLOM results, this thesis proposes and evaluates two CLOM processes that incorporate the AOLT concept. An AOLT (appropriate ontology label translation) result in the context of CLOM is a translation that is most likely to ensure the success of the subsequent MOM step. It has been shown through the evaluation of the AOLT-enabled CLOM process that translations are central to the improvement of mapping quality when incorporating MT and MOM techniques. Limitations of this research are discussed and future directions are suggested. Cross-lingual ontology mapping is a relatively unexplored area compared to monolingual ontology mapping, this work is among the initial efforts in this research field. It is the author's opinion that this thesis presents a concrete contribution to the field of cross-lingual ontology mapping.

BIBLIOGRAPHY

Alonso et al., 2005	Alonso L. S., Bas L. J., Bellido S., Contreras J., Benjamins R., Gomez M. J., WP10: Case Study eBanking D10.7 Financial Ontology, Data, Information and Process Integration with Semantic Web Services, FP6-507483, 2005
An et al., 2006	An Y., Borgida A., Mylopoulos J., Discovering the Semantics of Relational Tables Through Mappings. Journal on Data Semantics. VII:1-32, 2006
Bach et al., 2004	Bach T.L., Dieng-Kuntz R., Gandon F., On Ontology Matching Problems. In Proceedings of the 6th International Conference on Enterprise Information systems, pp. 236-243, 2004
Bateman, 1993	Bateman J. A., Ontology Construction and Natural Language. Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation. Ladseb-CNR Internal Report 01/93; editors: N. Guarino and R. Poli., 1993
Berlin & Motro, 2002	Berlin J., Motro A., Database Schema Matching Using Machine Learning with Feature Selection. In Proceedings of the 14th International Conference on Advanced Information systems Engineering, LNCS 2348, pp. 452-466, 2002
Berners-Lee et al., 2001	Berners-Lee T., Hendler J., Lassila O., The Semantic Web. Scientific American, May 2001, pp. 29-37, 2001
Bilke & Naumann, 2005	Bilke A., Naumann F., Schema Matching Using Duplicates. In Proceedings of the 21st International Conference on Data Engineering, pp. 69-80, 2005
Bouma, 2009	Bouma G., Cross-lingual Dutch to English Alignment Using EuroWordNet and Dutch Wikipedia. In Proceedings of the 4th International Workshop on Ontology Matching, CEUR-WS Vol-551, pp. 224-229, 2009

Bouquet et al., 2003	Bouquet P., Magnini B., Serafini L., Zanobini S., A SAT-based Algorithm for Context Matching. In Proceedings of the 4th International and Interdisciplinary Conference on Modeling and Using Context, LNCS 2680, pp. 66-79, 2003
Bunke & Kandel, 2000	Bunke H., Kandel A., Mean and Maximum Common Subgraph of Two Graphs. In Pattern Recognition Letters, 21, pp. 163-168, Elsevier, 2000
Caracciolo et al., 2007	Caracciolo C., Sini M., Keizer J., Requirements for the Treatment of Multilinguality in Ontologies within FAO, Food and Agricultural Organisation of the United Nations, 2007
Caracciolo et al., 2008	Caracciolo C., Euzenat J., Hollink L., Ichise R., Isaac A., Malaisé V., Meilicke C., Pane J., Shvaiko P., Stuckenschmidt H., Šváb-Zamazal O., Svátek V., Results of the Ontology Alignment Evaluation Initiative 2008. In Proceedings of the 3rd International Workshop on Ontology Matching, CEUR-WS Vol-431, pp. 73-119, 2008
Castano et al., 2000	Castano S., Antonellis V.D., Capitani di Vimercati S.D., Global Viewing of Heterogeneous Data Sources. IEEE Transactions on Knowledge and Data Engineering, 13(2): 277-297, 2000
Castano et al., 2006	Castano S., Ferrara A., Montanelli S., Matching Ontologies in Open Networked systems: Techniques and Applications. Journal on Data Semantics, V:25-63, 2006
Chalupsky, 2000	Chalupsky H., OntoMorph: A Translation System for Symbolic Knowledge. In Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, Colorado, USA, pp. 471-482, 2000
Chang & Lu, 2002	Chang C., Lu W., The Translation of Agricultural Multilingual Thesaurus, in Proceedings of the 3rd Asian Conference for Information Technology in Agriculture, 2002

Chang et al., 2005	Chang K., He B., Zhang Z., Toward Large Scale Integration: Building a Metaquerier over Databases on the Web. In Proceedings of the 2nd Biennial Conference on innovative Data Systems Research, pp. 44-55, 2005
Choi et al., 2006	Choi N., Song I., Han H., A Survey on Ontology Mapping. SIGMOD Record, Vol. 35, No.3 Sep. 2006, pp 34-41, 2006
Cimiano et al., 2010	Cimiano P., Montiel-Ponsoda E., Buitelaar P., Espinoza M., Gómez-Pérez A., A Note on Ontology Localization. Journal of Applied Ontology, vol.5 (2), pp.127-137, IOS Press, 2010
Clark et al., 2010	Clark A., Fox C., Lappin S., The Handbook of Computational Linguistics and Natural Language Processing. Wiley-Blackwell, July 2010. ISBN: 978-1-4051-5581-6
Clifton et al., 1997	Clifton C., Housman E., Rosenthal A., Experience with a Combined Approach to Attribute Matching Across Heterogeneous Databases. In Proceedings of the 7th IFIP Conference on Data Semantics, pp. 428-453, 1997
Compatangelo & Meisel, 2002	Compatangelo E., Meisel H., Intelligent Support to Knowledge Sharing Through the Articulation of Class Schemas. In Proceedings of the 6th International Conference on Knowledge-Based Intelligent Information & Engineering Systems, pp. 306-310, IOS Press, 2002.
Cui et al., 2004	Cui G., Chen F., Chen, H., Li S., OntoEdu: A Case Study of Ontology-based Education Grid System for E-learning, The Global Chinese Conference on Computers in Education conference, 2004
Dhamankar et al., 2004	Dhamankar R., Lee Y., Doan A., Halevy A., Domingos P., iMap: discovering Complex Semantic Matches between Database Schemas. In Proceedings of the 23rd International Conference on Management of Data, pp. 383-394, 2004

Do & Rahm, 2002 | Do H.H., Rahm E., COMA - A system for Flexible Combination of Schema Matching Approaches. In Proceedings of the 28th International Conference on Very Large Databases, pp. 610-621, 2002

Do et al., 2002 | Do H.H., Melnik S., Rahm E., Comparison of Schema Matching Evaluations. In Proceedings of Workshop on Web, Web-Services, and Database Systems, LNCS 2593, pp. 221-237, 2002

Doan et al., 2001 | Doan A., Domingos P., Halevy A., Reconciling Schemas of Disparate Data Sources: A Machine-learning Approach. In Proceedings of the 20th International Conference on Management of Data, pp. 509-520, 2001

Doan et al., 2002 | Doan A., Madhavan J., Domingos P., Halevy A., Learning to Map Between Ontologies on the Semantic Web. In Proceedings of the 11th International Conference on Wold Wild Web, ACM Press, pp. 662-673, 2002

Dou et al., 2004 | Dou D., McDermott D., Qi P., Ontology Translation on the Semantic Web. Journal on Data Semantics, II:35-57, 2004

Duan et al., 2010 | Duan S., Fokoue A., Srinivas K., One size does not fit all: Customizing Ontology Alignment using User Feedback. In Proceedings of the 9th International Semantic Web Conference, LNCS 6497, pp. 177-192, 2010

Ehrig, 2007 | Ehrig M., Ontology Alignment: Bridging the Semantic Gap. Semantic Web and Beyond: Computing for Human Experience. Springer, 2007

Ehrig & Euzenat, 2005 | Ehrig M., Euzenat J., Relaxed Precision and Recall for Ontology Matching. In Proceedings of the K-CAP Workshop on Integrating Ontologies, pp. 25-32, 2005

Ehrig & Staab, 2004 | Ehrig M., Staab S., QOM – Quick Ontology Mapping. In Proceedings of the 3rd International Semantic Web conference, LNCS 3298, pp. 683-697, 2004

Ehrig & Sure, 2004 — Ehrig M., Sure Y., Ontology Mapping – An Integrated Approach. In Proceedings of the 1st European Semantic Web Symposium, LNCS 3053, pp. 76-91, 2004

Espinoza et al., 2008 — Espinoza M., Gómez-Pérez A., Mena E., LabelTranslator - A Tool to Automatically Localize an Ontology. In Proceedings of the 5th European Semantic Web Conference, pp. 792-796, 2008

Euzenat, 1994 — Euzenat J., Brief Overview of T-tree: The Tropes Taxonomy Building Tool. In Proceedings of the 4th ASIS SIG/CR Workshop on Classification Research, pp. 69-87, 1994

Euzenat, 2004 — Euzenat J., Introduction to the EON Ontology Alignment Context. In Proceedings of the 3rd Intentional Workshop on Evaluation of Ontology-based Tools, 2004

Euzenat, 2007 — Euzenat J., Semantic Precision and Recall for Ontology Alignment Evaluation. In 20th International Joint Conference on Artificial Intelligence (IJCAI'07), pp. 348-353, 2007

Euzenat & Shvaiko, 2007 — Euzenat J., Shvaiko P., Ontology Matching. Springer Heidelberg 2007

Euzenat & Shvaiko, 2007a — Euzenat J., Shvaiko P., Overview of Matching Systems. In Ontology Matching, Springer Heidelberg, pp. 153-192, 2007

Euzenat & Shvaiko, 2008 — Euzenat J., Shvaiko P., Then Challenges for Ontology Matching. In Proceedings of the 7th International Conference on Ontologies, Databases and applications of Semantics, pp. 1164-1182, 2008

Euzenat & Valtchev, 2004 — Euzenat J., Valtchev P., Similarity-based Ontology Alignment in OWL-Lite. In Proceedings of the 15th European Conference on Artificial Intelligence, pp. 333-337, 2004

Euzenat et al., 2005 — Euzenat J., Stuckenschmidt H., Yatskevich M., Introduction to the Ontology Alignment Evaluation 2005. In Proceedings of the Workshop on Integrating Ontologies, CEUR-WS Vol-156, pp. 61-71, 2005

Euzenat et al., 2006	Euzenat J., Mochol M., Shvaiko P., Stuckenschmidt H., Šváb O., Svátek V., van Hage W. R., Yatskevich M., Results of the Ontology Alignment Evaluation Initiative 2006. In Proceedings of the International Workshop on Ontology Matching, CEUR-WS Vol-225, pp. 73-95, 2006
Euzenat et al., 2007	Euzenat J., Isaac A., Meilicke C., Shvaiko P., Stuckenschmidt H., Šváb O., Svátek V., van Hage W. R., Yatskevich M., First results of the Ontology Alignment Evaluation Initiative 2007. In Proceedings of the 2nd International Workshop on Ontology Matching, CEUR-WS Vol-304, pp. 96-132, 2007
Euzenat et al., 2009	Euzenat J., Ferrara A., Hollink L., Isaac A., Joslyn C., Malaisé V., Meilicke C., Nikolov A., Pane J., Sabou M., Scharffe F., Shvaiko P., Spiliopoulos V., Stuckenschmidt H., Šváb-Zamazal O., Svátek V., Trojahn C., Vouros G., Wang S., Results of the Ontology Alignment Evaluation Initiative 2009. In Proceedings of the 4th International Workshop on Ontology Matching, CEUR-WS Vol-551, pp. 73-126, 2009
Euzenat et al., 2010	Euzenat J., Ferrara A., Meilicke C., Nikolov A., Pane J., Scharffe F., Shvaiko P., Stuckenschmidt H., Šváb-Zamazal O., Svátek V., Trojahn C., Results of the Ontology Alignment Evaluation Initiative 2010. In Proceedings of the 5th International Workshop on Ontology Matching, CEUR-WS Vol-689, pp. 85-117, 2010
Falconer & Storey, 2007	Falconer S., Storey M.A., A Cognitive Support Framework for Ontology Mapping. In Proceedings of ISWC/ASWC, pp. 114-127, 2007
Fang et al., 2006	Fang K., Chang C., Chi Y., Leveraging Ontology-Based Traditional Chinese Medicine Knowledge System: Using Formal Concept Analysis, in Proceedings of the 9th Joint Conference on Information Sciences, 2006

Fisher, 1958	Fisher R.A., Statistical Methods for Research Workers. 13th edition, Hafner, 1958
Fu et al., 2009	Fu B., Brennan R., O'Sullivan D., Cross-Lingual Ontology Mapping – An Investigation of the Impact of Machine Translation. In Proceedings of the 4th Asian Semantic Web Conference, LNCS 5926, pp. 1-15, 2009
Fürst & Trichet, 2005	Fürst F, Trichet F, Axiom-based Ontology Matching: A Method and An Experiment. Technical report, Laboratoire d'Informatique de Nantes Atlantique, 2005
Giunchiglia & Shvaiko, 2003	Giunchiglia F., Shvaiko P., Semantic Matching. The Knowledge Engineering Review, 18(3): 265-280, 2003
Giunchiglia et al., 2007	Giunchiglia F., Yatskevich M., Shvaiko P., Semantic Matching: Algorithms and Implementation. Journal on Data Semantics, vol.IX, pp.1-38, 2007
Gruber, 1993	Gruber T., A Translation Approach to Portable Ontologies. Knowledge Acquisition 5(2):199-220, 1993
He et al., 2005	He H., Meng W., Yu C., Wu Z., WISE-Integrator: A System for Extracting and Integrating Complex Web Search interfaces of the Deep Web. In Proceedings of the 31st International Conference on Very Large Data Bases, pp. 1314-1317, 2005
Hollink et al., 2008	Hollink L., Van Assem M., Wang S., Isaac A., Schreiber G., Two Variations on Ontology Alignment Evaluation: Methodological Issues. In the 5th European Semantic Web Conference, LNCS 5021, pp. 388-401, 2008
Horridge et al., 2004	Horridge M., Knublauch H., Rector A., Stevens R., Wroe C., A Practical Guide to Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0. University of Manchester, 2004
Hovy, 1998	Hovy E., Combining and Standardising Large-scale, Practical Ontologies for Machine Translation and other uses. In Proceedings of the 1st International Conference on Language Resources and Evaluation, pp. 535-542, 1998

Hu & Qu, 2008	Hu W., Qu Y., Falcon-AO: A Practical Ontology Matching System. Web Semantics, 6, 3, pp. 237-239, 2008
Hutchins, 2004a	Hutchins J., Machine Translation: General Overview. Chapter 27 of Mitkov R. (Ed.) The Oxford Handbook of Computational Linguistics, Oxford, pp. 501-511, 2004
Hutchins, 2004b	Hutchins J., Machine Translation: General Overview. Chapter 28 of Mitkov R. (Ed.) The Oxford Handbook of Computational Linguistics, Oxford, pp. 512-528, 2004
Ichise, 2009	Ichise R., Evaluation of Similarity Measures for Ontology Matching. In Proceedings of the 22nd Annual Conference of the Japanese Society for Artificial Intelligence, LNAI 5447, pp. 15-25, 2009
Ichise et al., 2001	Ichise R., Takeda H., Honiden S., Rule Induction for Concept Hierarchy Alignment. In Proceedings of the 2nd Workshop on Ontology Learning, CEUR Proceedings Vol.38, 2001
Ichise et al, 2003	Ichise R., Takeda H., Honiden S., Integrating Multiple Internet Directories by Instance-based Learning. In Proceedings of the 18th International Joint conference on Artificial Intelligence, pp. 22-30, 2003
Ide & Véronis, 1998	Ide N., Véronis J., Word Sense Disambiguation: The State of the Art. Computation Linguistics, 1998 24 (1)
Jung, 2011	Jung J.J., Exploiting Multi-agent Platform for Indirect Alignment Between Multilingual Ontologies: A Case Study on Tourism Business. Expert Systems with Applications 38, pp. 5774-5780, 2011
Jung et al., 2009	Jung J. J., Håkansson A., Hartung R., Indirect Alignment between Multilingual Ontologies: A Case Study of Korean and Swedish Ontologies. In Proceedings of the 3rd International KES Symposium on Agents and Multi-agent Systems – Technologies and Applications, LNAI 5559, pp. 233-241, 2009

Kang & Naughton, 2003	Kang J., Naughton J., On Schema Matching with Opaque Column Names and Data Values. In Proceedings of the 22nd International Conference on Management of Data, pp. 205-216, 2003
Kalfoglou & Hu, 2005	Kalfoglou, Y., Hu, B.: CROSI Mapping System (CMS) - Result of the 2005 Ontology Alignment Contest. In Proceedings of Workshop on Integrating Ontologies, pp. 77-84, 2005
Kalfoglou & Schorlemmer, 2002	Kalfoglou Y., Schorlemmer M., Information-Flow-Based Ontology Mapping. In On the Move to Meaningful Internet Systems - DOA/CoopIS/ ODBASE Confederated International Conferences DOA, CoopIS and ODBASE 2002, Robert Meersman and Zahir Tari (Eds.). Springer-Verlag, London, UK, pp. 1132-1151, 2002
Kalfoglou & Schorlemmer, 2003	Kalfoglou Y, Schorlemmer M., Ontology Mapping: the State of the Art. The Knowledge Engineering Review, Vol. 18:1, pp.1-31, 2003
Kaza & Chen, 2007	Kaza S., Chen H., Evaluating Ontology Mapping Techniques: An Experiment in Public Safety Information Sharing. Decision Support Systems 45, 714-728, 2007
Kim et al., 2005	Kim J., Jang M., Ha Y.G., Sohn J.C., Lee S.J., MoA: OWL Ontology Merging and Alignment Tool for the Semantic Web. In Proceedings of the 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, LNCS 3533, pp. 722-731, 2005
Koehn, 2005	Koehn P., Europarl: A Parallel Corpus for Statistical Machine Translation, MT Summit, 2005
Koehn et al., 2003	Koehn, P., Och F., Marcu D., Statistical Phrase-based Translation. In HLT-NAACL: Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Edmonton, AL, Canada, pp. 127-133, 2003

Kolli & Doshi, 2008	Kolli R., Doshi P., Optima: Tool for ontology alignment with application to semantic reconciliation of sensor metadata for publication in sensormap. ICSC, pp. 484-485, 2008.
Kotis et al., 2006	Kotis K., Vouros G., Stergiou K., Towards Automatic Merging of Domain Ontologies: The HCONE-merge Approach. Journal of Web Semantics, 4(1): 60-79, 2006
Lacher & Groh, 2001	Lacher M.S., Groh G., Facilitating the Exchange of Explicit Knowledge through Ontology Mappings. In Proceedings of the 14th International Florida Artificial Intelligence Research Society Conference, Ingrid Russell and John F. Kolen (Eds.). AAAI Press, pp. 305-309, 2001
Lauser et al., 2002	Lauser B., Wildemann T., Poulos A., Fisseha F., Keizer J., Katz S., A Comprehensive Framework for Building Multilingual Domain Ontologies: Creating a Prototype Biosecurity Ontology. In Proceedings of DC-2002, pp. 113-123, 2002
Lee et al., 2002	Lee M.L., Yang L.H., Hsu W., Yang X., XClust: Clustering XML Schemas for Effective Integration. In Proceedings of the 11th International Conference on Information and Knowledge Management, pp. 292-299, 2002
Lerner, 2000	Lerner B.S., A Model for Compound Type Changes Encountered in Schema Evolution. ACM Transactions on Database Systems, 25(1): 83-127, 2000
Levenshtein, 1966	Levenshtein V.I., Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics-Doklady, vol.10, No.8, pp. 707-710, 1966
Li & Clifton, 2000	Li W., Clifton C., SEMINT: A Tool for Identifying Attribute Correspondences in Heterogeneous Databases using Neural Networks. Data and Knowledge Engineering, 33(1):49-84, 2000

Liang et al., 2005	Liang A., Sini M., Chang C., Li S., Lu W., He C., Keizer J., The Mapping Schema from Chinese Agricultural Thesaurus to AGROVOC, 6th Agricultural Ontology Service (AOS) Workshop on Ontologies: the more practical issues and experiences, 2005
Liang & Sini, 2006	Liang A., Sini M., Mapping AGROVOC and the Chinese Agricultural Thesaurus: Definitions, tools, procedures, New Review of Hypermedia and Multimedia, pp. 51-62, 12 (1), 2006
Macklovitch et al., 2000	Macklovitch E., Simard M., Langlais P., TransSearch: A Free Translation Memory on the World Wide Web. In Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000), pp. 1201-1208, 2000
Madhavan et al., 2001	Madhavan J., Bernstein P., Rahm E., Generic Schema Matching with Cupid. In Proceedings of the 27th International Conference on Very Large Databases, pp. 48-58, 2001
Madhavan et al., 2005	Madhavan, J., Bernstein P., Doan A., Halevy A., Corpus-based Schema Matching. In Proceedings of the 21st International Conference on Data Engineering, pp. 57-68, 2005
Manning et al., 2008	Manning C. D., Raghavan P., Schütze H., Introduction to Information Retrieval, Cambridge University Press. ISBN-13978-0-521-86571-5, 2008
Marcu & Wong, 2002	Marcu D., Wong W., A Phrase-based Joint Probability Model for Statistical Machine Translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp.133-139, 2002
McGuiness et al., 2000	McGuinness D.L., Fikes R., Rice J., Wilder S., The Chimaera ontology Environment. In Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, AAAI Press, pp. 1123-1124, 2000

Melnik et al., 2002 Melnik S., Garcia-Molina H., Rahm E., Similarity Flooding: A Versatile Graph Matching Algorithm. In Proceedings of the 18th International Conference on Data Engineering, pp. 117-128, 2002

Miller et al., 2001 Miller R., Hernandez M., Haas, L., Yan L., Ho H., Fagin F., Popa L., The Clio Project: Managing Heterogeneity. ACM SIGMOD Record, 30(1): 78-83, 2001

Milo & Zohar, 1998 Milo T., Zohar S., Using Schema Matching to Simplify Heterogeneous Data Translation. In Proceedings of the 24th International Conference on Very Large Databases, pp. 122-133, 1998

Minitab StatGuide, 2007 Minitab 15 StatGuide, Minitab Inc. 2007

Mitra et al., 1999 Mitra P., Wiederhold G., Jannink J., Semi-automatic Integration of Knowledge Sources. In Proceedings of the 2nd International Conference on Information Fusion, pp. 572-581, 1999

Mitra & Wiederhold, 2002 Mitra P., Wiederhold G., Resolving Terminological Heterogeneity in Ontologies. Workshop on Ontologies and Semantic Interoperability, 2002

Mitra et al., 2005 Mitra P., Noy N., Jaiswal A., Ontology Mapping discovery with Uncertainty. In Proceedings of the 4th International Semantic Web Conference, LNCS 3729, pp. 537-547, 2005

Modica et al., 2001 Modica G., Gal A., Jamil H., The Use of Machine-generated Ontologies in Dynamic Information Seeking. In Proceedings of the 9th International Conference on Cooperative Information Systems, LNCS 2172, pp. 433-448, 2001

Nagy et al., 2009 Nagy M., Vargas-Vera M., Stolarski P., DSSim Results for OAEI 2009. In Proceedings of the 4th International Workshop on Ontology Matching, CEUR-WS Vol-551, pp. 160-169, 2009

Navigli, 2009 Roberto Navigli, Word sense disambiguation: A survey. ACM Computing Surveys, 41, 2, Article 10, DOI=10.1145/1459352.1459355, 2009

Nerbonne et al., 1999 — Nerbonne J., Heeringa W., Kleiweg P., Edit Distance and Dialect Proximity. In: Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison, 2nd ed. CSLI, Stanford, v-xv, 1999

Ngai et al., 2002 — Ngai G., Carpuat M., Fung P., Identifying Concepts Across Languages: A First Step towards A Corpus-based Approach to Automatic Ontology Alignment. In Proceedings of the 19th International Conference on Computational Linguistics, vol.1, pp. 1-7, 2002

Niles & Pease, 2001 — Niles I., Pease A., Towards A Standard Upper Ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems, pp. 2-9, 2001

Nirenburg & Raskin, 2001 — Nirenburg S. & Raskin V., Ontological Semantics, Formal Ontology, and Ambiguity. Proceedings of the 2nd International Conference on Formal Ontology in Information Systems. ACM Press, pp. 151-161, 2001

Niwa et al., 1997 — Niwa Y., Nishioka S., Iwayama M., Takano A., Topic Graph Generation for Query Navigation. In Proceedings of Natural Language Processing Pacific Rim Symposium, pp. 95–100, 1997

Nottelmann & Straccia, 2005 — Nottelmann H., Straccia U., sPLMap: A Probabilistic Approach to Schema Matching. In Proceedings of the 27th European Conference on Information Retrieval Research, pp. 81-95, 2005

Noy & McGuinness, 2001 — Noy N. F., McGuinness D. L., Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001

Noy & Musen, 1999 — Noy N.F., Musen M.A., SMART: Automated Support for Ontology Merging and Alignment. In Proceedings of the 12th Banff Workshop on knowledge Acquisition, Modeling and Management, 1999

Noy & Musen, 2000	Noy N.F, Musen M.A., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In Proceedings of the National Conference on Artificial Intelligence, 2000
Noy & Musen, 2001	Noy N.V., Musen M.A., Anchor-PROMPT: Using Non-Local Context for Semantic Matching. In Proceedings of the Workshop on Ontologies and Information Sharing, pp. 63-70, 2001
Noy & Musen, 2002a	Noy N.V., Musen M.A., PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. In the 18th National Conference on Artificial Intelligence, pp. 744-750, 2002
Noy & Musen, 2002b	Noy N.F., Musen M.A., Evaluating Ontology-Mapping Tools: Requirements and Experience. OntoWeb-SIG3 Workshop, Siguenza, Spain, 2002
Och & Ney, 2000	Och F.J., Ney H., Improved Statistical Alignment Models. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, 2000
O'Sullivan et al., 2007	O'Sullivan D., Wade V., Lewis D., Understanding as We Roam. IEEE Internet Computing, 11(2), pp. 26-33, 2007
Pan et al., 2005	Pan R., Ding Z., Yu Y., Peng Y., A Bayesian Network Approach to Ontology Mapping. In Proceedings of the 3rd International Semantic Web Conference, LNCS 3298, pp. 563-577, 2005
Papineni et al., 2002	Papineni K., Roukos S., Ward T., Zhu W.J., BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311-318, 2002
Pazienta & Stellato, 2005	Pazienta M., Stellato A., Linguistically Motivated Ontology Mapping for the Semantic Web. In Proceedings of the 2nd Italian Semantic Web Workshop, pp. 14-16, 2005

Pazienza & Stellato, 2006a	Pazienza M. T., Stellato A., An Open and Scalable Framework for Enriching Ontologies with Natural Language Content. In Proceedings of the 19th International Conference on Industrial, Engineering and other Applications of Applied Intelligent Systems (IEA/AIE 2006). Springer, LNCS 4031, pp. 990-999, 2006
Pazienza & Stellato, 2006b	Pazienza M. T., Stellato A., Exploiting Linguistic Resources for Building Linguistically Motivated Ontologies in the Semantic Web. In Proceedings of OntoLex Workshop 2006: Interfacing Ontologies and Lexical Resources for Semantic Web Technologies, 2006
Palopoli et al., 2003	Palopoli L, Terracina G., Ursino D., DIKE : A system Supporting the Semi-automatic Construction of Cooperative Information Systems from Heterogeneous Databases. Software-Practice and Experience, 33(9), pp. 847-884, 2003
Prasad et al., 2002	Prasad S., Peng Y., Finin T., Using Explicit Information to Map Between Two Ontologies. In Proceedings of the AAMAS 2002 Workshop on Ontologies in Agent Systems, pp. 52-57, 2002
Rahm et al., 2004	Rahm E., Do H., Maβmann S., Matching Large XML Schemas. ACM SIGMOD Record, 33(4):26-31, 2004
Ruthven & Lalmas, 2003	Ruthven I., Lalmas M., A Survey On the Use of Relevance Feedback for Information Access Systems. Knowledge Engineering Review 18, 2, pp. 95-145, 2003
Shimoji & Wada, 2008	Shimoji Y., Wada T., Hirokawa S., Dynamic Thesaurus Construction from English-Japanese Ditionary. In Proceedings of the 2nd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS 2008). IEEE Computer Society, pp. 918-923, 2008

Shvaiko & Euzenat, 2005	Shvaiko P., Euzenat J., A Survey of Schema-based Matching Approaches. Journal on Data Semantics, vol.IV, pp. 146-171, 2005
Shvaiko & Euzenat, 2008	Shvaiko P., Euzenat J., Ten Challenges for Ontology Matching. In Proceedings of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics, pp. 1164-1182, 2008
Sosnovsky & Gavrilova, 2006	Sosnovsky S., Gavrilova T., Development of Educational Ontology for C-programming, International Journal "Information Theories and Applications" Volume 13, pp. 303 – 308, 2006
Srinivasan, 1992	Srinivasan P., Thesaurus Construction. Data Structures and Algorithms, Prentice-Hall, 1992
Straccia & Troncy, 2005	Straccia U., Troncy R., oMAP: Combining Classifiers for Alignment Automatically OWL Ontologies. In Proceedings of the 6th International Conference on Web Information Systems Engineering, pp. 133-147, 2005
Stumme & Maedche, 2001	Stumme G., Maedche A., FCA-MERGE: Bottom-up Merging of Ontologies. In Proceedings of IJCAI, pp. 225-230, 2001
Suárez-Figueroa & Gómez-Pérez, 2008	Suárez-Figueroa, M., Gómez-Pérez, A., First Attempt Towards A Standard Glossary of Ontology Engineering Terminology. In Proceedings of the 8th International Conference on Terminology and Knowledge Engineering, 2008.
Tang et al., 2006	Tang J., Li J., Liang B., Huang X., Li Y., Wang K., Using Bayesian Decision for Ontology Mapping. Journal of Web Semantics, 4(1):243-262, 2006

Tenenbaum et al., 2011 — Tenenbaum J.D., Whetzel P.L., Anderson K., Borromeo C.D., Dinov I.D., Gabriel D., Kirschner B., Mirel B., Morris T., Noy N., Nyulas C., Rubenson D., Saxman P.R., Singh H., Whelan N., Wright Z., Athey B.D., Becich M.J., Ginsburg G.S., Musen M.A., Smith K.A., Tarantal A.F., Rubin D.L., Lyster P., The Biomedical Resource Ontology (BRO) to Enable Resource Discovery in Clinical and Translational Research. Journal of Biomedical Informatics, Journal of Biomedical Informatics, 44, 1, pp. 137-145, 2011

Trojahn, 2008 — Trojahn C., Quaresma P., Vieira R., A Framework for Multilingual Ontology Mapping. In Proceedings of the 6th edition of the Language Resources and Evaluation Conference, pp. 1034-1037, 2008

Trojahn, 2010 — Trojahn C., Quaresma P., Vieira R., An API for Multi-lingual Ontology Matching. In Proceedings of the 7th Conference on International Language Resources and Evaluation, ISBN 2-9517408-6-7, pp. 3830-3835, 2010

van Rigsbergen, 1975 — Van Rijsbergen C.F., Information Retrieval. Butterworths, London (UK), 1975

Vas, 2007 — Vas R., Educational Ontology and Knowledge Testing, The Electronic Journal of Knowledge Management of Volume 5 Issue 1, pp. 123 – 130, 2007

Velegrakis et al., 2004 — Velegrakis Y., Miller R., Ropa L., Muylopoulos J., ToMAS: A system for Adapting Mappings While Schemas Evolve. In Proceedings of the 20th International Conference on Data Engineering, pp. 862, 2004

Wang et al., 2004 — Wang J., Wen J., Lochovsky F., Ma W., Instance –based Schema Matching for Web Databases by Domain-specific Query Probing. In Proceedings of the 30th International Conference on Very Large Data Bases, pp. 408-419, 2004

Wang et al., 2009	Wang S., Isaac A., Schopman B., Schlobach S., Van der Meij L., Matching Multi-lingual Subject Vocabularies. In Proceedings of the 13th European Conference on Digital Libraries, LNCS 5714, pp. 125-137, 2009
Wang et al., 2010	Wang Z., Zhang X., Hou L., Zhao Y., Li J., Qi Y., Tang J., RiMOM Results for OAEI 2010. In Proceedings of the 5th International Workshop on Ontology Matching, CEUR-WS Vol-689, pp. 195-202, 2010
Way, 2010	Way A., Machine Translation. In A. Clark, C. Fox and S. Lappin (eds.) The Handbook of Computational Linguistics and Natural Language Processing, Wiley Blackwell, Chichester, UK, pp. 531-573, 2010
Weaver & Wiener, 1947	Weaver W. and Wiener N. Correspondence, 1947, at http://www.mt-archive.info/Weaver-1947-original.pdf last retrieved in April 2011
Weiss, 2010	Weiss N.A., Introductory Statistics, 9th Edition. Pearson, 2010
Xu & Embley, 2003	Xu L., Embley D., Discovering Direct and Indirect Matches for Schema Elements. In Proceedings of the 8th International Conference on Database systems for Advanced Applications, pp. 39-46, 2003
Zhang et al., 2000	Zhang Z., Zhang C., Ong S. S., Building an Ontology for Financial Investment, in Intelligent Data Engineering and Automated Learning - IDEAL 2000, Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference, pp. 308-313, 2000
Zhang et al., 2008	Zhang X., Zhong Q., Li J., Tang J., Xie G., Li H., RiMOM Results for OAEI 2008. In Proceedings of the 3rd International Workshop on Ontology Matching, pp. 182-189, 2008

APPENDICES

The appendices present additional details for this thesis.

- Appendix A contains a table of content for the DVD submitted along this thesis.
- Appendix B presents a list of monolingual ontology matching tools mentioned in chapter 2.
- Appendix C contains code snippets of the CLOM systems presented in this thesis.
- Appendix D presents the document type definitions used in this thesis.
- Appendix E presents the scattered plots generated during the trial evaluations on confidence levels.

APPENDIX A. DVD CONTENT

This appendix contains a table of content for the DVD that companies this thesis. There are seven folders on the root directory of the DVD, including *Thesis*, *Baseline*, *SOCOM*, *SOCOM++*, *BaselineExperiments*, *SOCOMExperiments* and *SOCOM++Experiments*. Their content is discussed below.

Folder Name	Content
`root/Thesis`	Contains two files: - The `.doc` file contains a copy of this thesis in the Microsoft Word format. - The `.pdf` file contains a copy this thesis in the portable document format.
`root/Baseline`	Contains the source code used for the implementation of the baseline system (discussed in chapter 3).
`root/SOCOM`	Contains the source code used for the implementation of prototype one: SOCOM (discussed in chapter 4).
`root/SOCOM++`	Contains the source doe for prototype two: SOCOM++ (discussed in chapter 5).
`root/BaselineExperiments`	Contains the raw experimental data and evaluations of the baseline system (discussed in chapter 3). There are two folders: - The `root/BaselineExperiment/Experiment1` folder contains the raw data and evaluation results of the experiment shown in chapter 3, section 3.4.1. - The `root/BaselineExperiment/Experiment2` folder contains the raw data and evaluation results of the experiment shown in chapter 3, section 3.4.2.
`root/SOCOMExperiments`	Contains the raw experimental data and evaluation results of SOCOM (discussed in chapter 4). There are three folders: - The `root/SOCOMExperiments/Experiment1` folder contains the raw data and evaluation results of the experiment shown in chapter 4, section 4.5.1. - The `root/SOCOMExperiments/Experimient2` folder contains the raw data and evaluation results of the experiment shown in chapter 4, section 4.5.2. - The `root/SOCOMExperiments/CaseStudy` folder contains the ontologies and the mappings generated in the case study discussed in chapter 4, section 4.6.

`root/SOCOM++Experiments`	Contains the raw experimental data and evaluation results of SOCOM++ (discussed in chapter 5). There are six folders: - The `root/SOCOM++Experiments/TrialOne` folder contains the raw data and evaluation results of the experiment shown in chapter 5, section 5.4.2.1. - The `root/SOCOM++Experiments/TrialTwo` folder contains the raw data and evaluation results of the experiment shown in chapter 5, section 5.4.2.2. - The `root/SOCOM++Experiments/TrialThree` folder contains the raw data and evaluation results of the experiment shown in chapter 5, section 5.4.2.3. - The `root/SOCOM++Experiments/TrialFour` folder contains the raw data and evaluation results of the experiment shown in chapter 5, section 5.4.3.1. - The `root/SOCOM++Experiments/TrialFive` folder contains the raw data and evaluation results of the experiment shown in chapter 5, section 5.4.3.2. - The `root/SOCOM++Experiments/TrialSix` folder contains the raw data and evaluation results of the experiment shown in chapter 5, section 5.4.3.3.

APPENDIX B. MONOLINGUAL MATCHING TOOLS

This Appendix contains a list of matching tools mentioned in chapter 2, section 2.5.

The matching tools surveyed by Choi et al. [Choi et al., 2006] include SMART [Noy & Musen, 1999], PROMPT [Noy & Musen, 2000], OntoMorph [Chalupsky, 2000], HICAL (HIerarchical Concept ALignment system) [Ichise et al., 2001], Anchor-PROMPT [Noy & Musen, 2001], CROSI [Kalfoglou & Hu, 2005], FCA-Merge [Stumme & Maedche, 2001] and CHIMAERA [McGuiness et al., 2000].

The matching tools surveyed by Kalfoglou & Schorlemmer [Kalfoglou & Schorlemmer, 2003] include FCA-Merge [Stumme & Maedche, 2001], IF-Map [Kalfoglou & Schorlemmer, 2002], SMART [Noy & Musen, 1999], PROMPT [Noy & Musen, 2001], PROMPDIFF [Noy & Musen, 2002a], CHIMAERA [McGuiness et al., 2000], GLUE [Doan et al., 2002], CAIMAN [Lacher & Groh, 2001], ITTalks [Prasad et al., 2002], ONION [Mitra & Wiederhold, 2002] and ConcepTool [Compatangelo & Meisel, 2002].

The matching tools surveyed by Euzenat & Shvaiko [Euzenat & Shvaiko, 2007 p.153] include DELTA [Clifton et al., 1997], Hovy [Hovy, 1998], TranScm [Milo & Zohar, 1998], DIKE [Palopoli et al., 2003], SKAT [Mitra et al., 1999], ONION [Mitra & Wiederhold, 2002], Artemis [Castano et al., 2000], H-Match [Castano et al., 2006], Tess [Lerner, 2000], Anchor-PROMPT [Noy & Musen, 2001], OntoBuilder [Modica et al., 2001], Cupid [Madhavan et al., 2001], COMA/COMA++ [Do & Rahm, 2002], Similarity flooding [Melnik et al., 2002], XClust [Lee et al., 2002], ToMAS [Velegrakis et al., 2004], MapOnto [An et al., 2006], OntoMerge [Dou et al., 2004], CtxMatch [Bouquet et al., 2003], S-Match [Giunchiglia & Shvaiko, 2003], HCONE [Kotis et al., 2006], MoA [Kim et al., 2005], ASCO [Bach et al., 2004], BayesOWL [Pan et al., 2005], OMEN [Mitra et al., 2005], DCM [Chang et al., 2005], T-tree [Euzenat, 1994], CAIMAN [Lacher & Groh, 2001], FCA-merge [Stumme & Maedche, 2001], LSD [Doan et al., 2001], GLUE [Doan et al., 2002], iMap [Dhamankar at al., 2004], Automatch [Berlin & Motro, 2002], SBI & NB [Ichise et al., 2003], Kang & Naughton [Kang & Naughton, 2003], Dumas [Bilke & Naumann, 2005], Wang et al. [Wang et al., 2004], sPLMap [Nottelmann & Straccia, 2005], SEMINT [Li & Clifton, 2000], Clio [Miller et al., 2001], IF-Map [Kalfoglou & Schorlemmer, 2002], NOM [Ehrig & Sure, 2004], QOM [Ehrig & Staab, 2004], oMap [Straccia & Troncy, 2005], Xu & Embley [Xu & Embley, 2003], Wise-Integrator [He et al., 2005], OLA [Euzenat & Valtchev, 2004], Falcon-AO [Hu & Qu, 2008], RiMOM [Tang et al., 2006] and Corpus-based matching [Madhavan et al., 2005].

APPENDIX C. CODE SNIPPETS

C.1. Appendix Overview

This appendix presents some of the code that has been implemented during this Ph.D.. For a complete record, see the DVD which accompanies this thesis. Section C.2 includes code snippets for the implementation of the baseline system (discussed in chapter 3). Section C.3 contains code snippets used for the implementation of the SOCOM prototype (discussed in chapter 4). Section C.4 presents code snippets used for the implementation of the SOCOM++ prototype (discussed in chapter 5).

C.2. The Baseline System Code Snippets

Figure C-1 presents a code snippet that demonstrates how an OntModel is created.

```
    public OntModel copy() {
        setProxy();
        String source = "http://annotation.semanticweb.org/iswc/iswc.owl";
        OntModel originalOnt = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM, null);
        originalOnt.getDocumentManager().addAltEntry("http://annotation.semanticweb.org/iswc/iswc.owl",
                "file://localhost/Users/BoFu/Documents/PHD/NGL/MatchingOntologies/iswc.owl");
        originalOnt.read(source);
        ontologyCopy = ModelFactory.createOntologyModel(ProfileRegistry.OWL_LANG);
        BasicConfigurator.configure();
        translatedTerms = new ArrayList();
        counter = 0;
        for (Iterator i = originalOnt.listClasses(); i.hasNext();) {
            describeClass(System.out, (OntClass) i.next());
        }
        try {
            writeToFile(ontologyName + ".owl");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        return ontologyCopy;
    }
```

Figure C-1. OntModel Creation

Figure C-2 presents a code snippet demonstrating how concatenated class labels (using underscores) are converted to their natural language formats. In addition, it shows how these labels are translated next using the GoogleTranslate API and finally new classes are created using the OntModel.

```java
            //replace underscores with white spaces
            finalSplitted = afterHash.replace("_", " ");
            try {
                translatedAfterHash = Translate.translate(finalSplitted, Language.ENGLISH, Language.CHINESE_SIMPLIFIED);
                if (translatedTerms.contains(translatedAfterHash) && rootClass) {
                    if (cls.isUnionClass()) {
                        translatedCls = ontologyCopy.createUnionClass(baseNs + translatedAfterHash + counter++, null);
                    } else if (cls.isIntersectionClass()) {
                        translatedCls = ontologyCopy.createIntersectionClass(baseNs + translatedAfterHash + counter++, null);
                    } else if (cls.isComplementClass()) {
                        translatedCls = ontologyCopy.createComplementClass(baseNs + translatedAfterHash + counter++, null);
                    } else {
                        translatedCls = ontologyCopy.createClass(baseNs + translatedAfterHash + counter++);
                    }
                } else if (!translatedTerms.contains(translatedAfterHash) && rootClass) {
                    if (cls.isUnionClass()) {
                        translatedCls = ontologyCopy.createUnionClass(baseNs + translatedAfterHash, null);
                    } else if (cls.isIntersectionClass()) {
                        translatedCls = ontologyCopy.createIntersectionClass(baseNs + translatedAfterHash, null);
                    } else if (cls.isComplementClass()) {
                        translatedCls = ontologyCopy.createComplementClass(baseNs + translatedAfterHash, null);
                    } else {
                        translatedCls = ontologyCopy.createClass(baseNs + translatedAfterHash);
                    }
                    translatedTerms.add(translatedAfterHash);
                } else {
                    if (cls.isUnionClass()) {
                        translatedCls = ontologyCopy.createUnionClass(baseNs + translatedAfterHash, null);
                    } else if (cls.isIntersectionClass()) {
                        translatedCls = ontologyCopy.createIntersectionClass(baseNs + translatedAfterHash, null);
                    } else if (cls.isComplementClass()) {
                        translatedCls = ontologyCopy.createComplementClass(baseNs + translatedAfterHash, null);
                    } else {
                        translatedCls = ontologyCopy.createClass(baseNs + translatedAfterHash);
                    }
                }
            } catch (Exception ex) {
```

Figure C-2. OntClass Creation in O_1'

Figure C-3 demonstrates how concatenated datatype property labels (using capital letters) are converted to their natural language formats. Also, how they are translated next to create new datatype properties using the OntModel.

```java
            //decompose concatenated words
            char[] c = afterHash.toCharArray();
            for (int x = 0; x < c.length; x++) {
                int i = c[x];
                if (Character.isUpperCase(c[x])) {
                    splitted = splitted + afterHash.substring(0, afterHash.indexOf(i)) + " ";
                    afterHash = afterHash.substring(afterHash.indexOf(i));
                }
            }
            finalSplitted = (splitted + afterHash).trim();
            try {
                translatedAfterHash = Translate.translate(finalSplitted, Language.ENGLISH, Language.CHINESE_SIMPLIFIED);
                translatedAfterHash = translatedAfterHash + "_??";
                if (prop.isDatatypeProperty() && prop.isFunctionalProperty()) {
                    translatedProperty = ontologyCopy.createDatatypeProperty(baseNs + translatedAfterHash).asFunctionalProperty();
                } else if (prop.isObjectProperty() && prop.isFunctionalProperty()) {
                    translatedProperty = ontologyCopy.createObjectProperty(baseNs + translatedAfterHash).asFunctionalProperty();
                } else if (prop.isObjectProperty() && prop.isTransitiveProperty()) {
                    translatedProperty = ontologyCopy.createObjectProperty(baseNs + translatedAfterHash).asTransitiveProperty();
                } else if (prop.isObjectProperty() && prop.isSymmetricProperty()) {
                    translatedProperty = ontologyCopy.createObjectProperty(baseNs + translatedAfterHash).asSymmetricProperty();
                } else if (prop.isObjectProperty() && prop.isInverseFunctionalProperty()) {
                    translatedProperty = ontologyCopy.createObjectProperty(baseNs + translatedAfterHash).asInverseFunctionalProperty();
                } else if (prop.isDatatypeProperty()) {
                    translatedProperty = ontologyCopy.createDatatypeProperty(baseNs + translatedAfterHash);
                } else if (prop.isObjectProperty()) {
                    translatedProperty = ontologyCopy.createObjectProperty(baseNs + translatedAfterHash);
                }
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
    return translatedProperty;
}
```

Figure C-3. Datatype Property Creation in O_1'

Figure C-4 demonstrates how the Alignment API is used to generate matches in the baseline system.

```
        URI uri1 = new URI(sourceURL);
        URI uri2 = new URI(targetURL);
        Parameters p = new BasicParameters();
        OWLOntology source = loadOntology(uri1);
        OWLOntology target = loadOntology(uri2);
        AlignmentProcess a1 = new NameAndPropertyAlignment(source, target);
        a1.align((Alignment) null, p);
        File a1RDF = new File(resultsDirectory + nameAndPropertyAlignmentRDF);
        PrintWriter printWriterA1RDF = new PrintWriter(a1RDF);
        AlignmentVisitor alignVistA1RDF = new RDFRendererVisitor(printWriterA1RDF);
        a1.render(alignVistA1RDF);
        printWriterA1RDF.flush();
        printWriterA1RDF.close();
        a1RDF.createNewFile();
```

Figure C-4. Creating Matches with the Alignment API

C.3. Prototype One: SOCOM Code Snippets

The code snippet shown in figure C-5 loads a locally stored ontology, iterate through the classes from within and extract the class labels.

```
    static final String sourceDirectory = "C:\\Users\\BoFu\\Documents\\PHD\\NGL\\MatchingOntologies\\swrc_v0.3.owl";
    OntModel m = ModelFactory.createOntologyModel();
    DecompoundService ds = new DecompoundService();

    public Document createTranslationRepository(){
        doc.addContent(tr);
        return doc;
    }

    public void loadSourceOntology() throws FileNotFoundException {
        FileInputStream fis = new FileInputStream(sourceDirectory);
        m.read(fis, "http://swrc.ontoware.org/ontology#");
    }

    public String getClassLabel() throws Exception {
        createTranslationRepository();
        ExtendedIterator root = m.listNamedClasses();
        while (root.hasNext()) {
            classURI = root.next().toString();
            if (classURI.indexOf("#") > 0 && classURI.contains("Thing") == false) {
                classLabel = classURI.substring(classURI.indexOf("#") + 1);
                newClassLabel = ds.byCapital(classLabel);
                createClassElement(newClassLabel);
            }
        }
        return newClassLabel;
    }
```

Figure C-5. Iterating through the Classes in an Ontology

The code snippet in figure C-6 illustrates how the semantic surrounding of an ontology class is generated.

```java
public String[] generateClassSurrounding (String sourceClassTerm) throws Exception{
    OntClass sourceClass = m.getOntClass(namespace + sourceClassTerm);
    ArrayList<String> classSurroundingArray = new ArrayList<String>();
    classSurroundingArray.add(sourceClassTerm);
    if (sourceClass != null && sourceClass.listSubClasses() != null){
        List subClassList = sourceClass.listSubClasses().toList();
        for (int counter = 0; counter < subClassList.size(); counter++){
            String subLabel = ((OntClass)subClassList.get(counter)).getLocalName();
            if (subLabel != null){
                classSurroundingArray.add(subLabel);
            }
        }
    }
    if (sourceClass != null && sourceClass.listSuperClasses() != null){
        List superClassList = sourceClass.listSuperClasses().toList();
        for (int counter = 0; counter < superClassList.size(); counter++){
            String superLabel = ((OntClass)superClassList.get(counter)).getLocalName();
            if (superLabel != null){
                classSurroundingArray.add(superLabel);
            }
        }
    }
    String[] returnArray = new String[classSurroundingArray.size()];

    for (int i =0; i < classSurroundingArray.size(); i++){
        returnArray[i] = classSurroundingArray.get(i);
    }
    return returnArray;
}
```

Figure C-6. Class Semantic Surrounding Generation

The code snippet in figure C-7 shows the generation of candidate translations for a source resource label via the GoogleTranslate API and the WindowsLive translator.

```java
public void createClassElement (String sourceLabel){
    Element resultTag = new Element("Result");
    Element sourceIDTag = new Element("SourceID").setText("SC" + (sourceID++));
    Element sourceValueTag = new Element("SourceValue").setText(sourceLabel);
    Element candidateCollectionTag = new Element("CandidateCollection");
    tr.addContent(resultTag);
    resultTag.addContent(sourceIDTag).addContent(sourceValueTag).addContent(candidateCollectionTag);
    try {
        addCandidate(sourceIDTag.getText(), ts.ChiToEngUsingGoogle(classLabel), candidateCollectionTag);
    } catch (Exception e) {
        e.printStackTrace();
    }
    try {
        addCandidate(sourceIDTag.getText(), ts.ChiToEngUsingWindows(classLabel), candidateCollectionTag);
    } catch (Exception e) {
        e.printStackTrace();
    }
    store();
}
public String EngToChiUsingGoogle(String sourceLabel) throws Exception {
    toTranslateLabel = ds.byCapital(sourceLabel);
    if (toTranslateLabel != "Thing") {
        googleTranslationResult = Translate.translate(toTranslateLabel, Language.ENGLISH, Language.CHINESE_SIMPLIFIED);
        System.out.println(googleTranslationResult);
    }
    return googleTranslationResult;
}
public String EngToChiUsingWindows(String sourceLabel) throws Exception {
    toTranslateLabel = ds.byCapital(sourceLabel);
    if (toTranslateLabel != "Thing") {
        WindowsLiveClient client = new WindowsLiveClient();
        windowsTranslaionResult = client.translate(toTranslateLabel, "en", "zh");
        System.out.println(windowsTranslaionResult);
    }
    return windowsTranslaionResult;
}
```

Figure C-7. Generating and Storing Candidate Translations

The code snippet in figure C-8 illustrates the generation of synonyms for an individual label using the WordNet thesaurus and the Dictionary.com API.

```java
    public void createIndividualElement (String targetLabel){
        Element resultTag = new Element("Result");
        Element targetIDTag = new Element("TargetID").setText("TI" + (targetID ++));
        Element targetValueTag = new Element("TargetValue").setText(targetLabel);
        Element synonymCollectionTag = new Element("SynonymCollection");
        tr.addContent(resultTag);
        resultTag.addContent(targetIDTag).addContent(targetValueTag).addContent(synonymCollectionTag);
        try {
            String[] allSynonyms = le.getSynonymFromWordNet(newIndividualLabel);
            for (int i = 0; i < allSynonyms.length; i++){
                addSynonym(targetIDTag.getText(), allSynonyms[i], synonymCollectionTag);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            String[] allSynonyms = le.getSynonymFromThesaurusDotCom(newIndividualLabel);
            for (int i = 0; i < allSynonyms.length; i++){
                addSynonym(targetIDTag.getText(), allSynonyms[i].trim(), synonymCollectionTag);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        store();
    }
```

```java
    public String[] getSynonymFromWordNet(String targetLabel){
        RiWordnet wordnet = new RiWordnet();
        String pos = wordnet.getBestPos(targetLabel);
        String[] synonymArray = wordnet.getAllSynonyms(targetLabel, pos, 10);
        return synonymArray;
    }
```

```java
    public String[] getSynonymFromThesaurusDotCom(String targetLabel) throws IOException, SAXException, ParserConfigurationException{
        String key = "18ic4e3ywiypobiwk1drsqjts690ohtqqc2vq5tnos";
        String website = "http://api-pub.dictionary.com/v001?vid=" + key + "&q=" + targetLabel + "&type=synonyms";
        setProxy();
        URL url = new URL(website);
        HttpURLConnection c = (HttpURLConnection) url.openConnection();
        BufferedReader br = new BufferedReader(new InputStreamReader(c.getInputStream()));
        StringBuffer sb = new StringBuffer();
        String line;
        while ((line = br.readLine()) != null){
            sb.append(line);
        }
        br.close();
        String result = sb.toString();
        DocumentBuilderFactory factory =  DocumentBuilderFactory.newInstance();
        InputSource source = new InputSource(new StringReader(result));
        Document doc = factory.newDocumentBuilder().parse(source);
        String totalSynonym = doc.getDocumentElement().getAttribute("totalresults");
        int total = Integer.parseInt(totalSynonym.trim());
        String[] synonyms = new String[total];
        NodeList allSynonyms = (NodeList) doc.getElementsByTagName("headword");
        if(allSynonyms!=null && ((org.w3c.dom.NodeList) allSynonyms).getLength()>0){
            for(int i=0; i<((org.w3c.dom.NodeList) allSynonyms).getLength(); i++){
                Element headwordEle = (Element) ((org.w3c.dom.NodeList) allSynonyms).item(i);
                String synonym = headwordEle.getTextContent();
                for (int j = 0; j < synonyms.length; j++) {
                    synonyms[j] = synonym;
                }
            }

        }
        return synonyms;
    }
```

Figure C-8. Synonym Generation for an Individual in O_2

The code snippet in figure C-9 illustrates the generation of keywords for a candidate translation from Wikipedia via the Yahoo Term Extraction tool.

```java
public String[] getKeywords (String query){
    ArrayList keywordsList = new ArrayList();
    keywordsList.add(0, query);
    final String request = "http://search.yahooapis.com/ContentAnalysisService/V1/termExtraction";
    HttpClient client = new HttpClient();
    PostMethod method = new PostMethod(request);
    client.getHostConfiguration().setProxy("www-proxy.cs.tcd.ie", 8080);
    try {
        method.addParameter("appid","rxcF6yHV34G2PJeWEkUI5TDpINOvM5uhdbkNbw8i0LyDXLBa3voUFY7SzKGiphU-");
        method.addParameter("query",query);
        String context = generateContextFromWikipedia(query);
        method.addParameter("context",context);
    } catch (IllegalArgumentException e1) {
        e1.printStackTrace();
    } catch (IOException e1) {
        e1.printStackTrace();
    }
    int statusCode = 0;
    try {
        statusCode = client.executeMethod(method);
    } catch (HttpException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    InputStream rstream = null;
    try {
        rstream = method.getResponseBodyAsStream();
    } catch (IOException e) {
        e.printStackTrace();
    }
    BufferedReader br = new BufferedReader(new InputStreamReader(rstream));
```

```java
public String generateContextFromWikipedia (String query) throws IOException{
    String all = null;
    setProxy();
    String wikipediaUrl = "http://en.wikipedia.org/w/index.php?title=" + query + "&amp";
    StringExtractor se = new StringExtractor(wikipediaUrl);
    String unquotedAll = null;
    try {
        all = se.extractStrings(false);
        if (all != null){
            unquotedAll = all.replaceAll("\"", "\\\\'");
        }
    } catch (ParserException e1) {
        e1.printStackTrace();
    }
    return unquotedAll;
}
```

Figure C-9. Keyword Generation for a Candidate Translation

The code snippet in figure C-10 demonstrates how a space/case-insensitive edit distance string comparison algorithm is implemented to compare two character strings (i.e. one label vs. another label).

```java
    public double getScore(String s1, String s2) {
        Distance<CharSequence> d = new CasePunctuationDistance();
        double score = d.distance(s1, s2);
        return score;
    }

    static class CasePunctuationDistance extends WeightedEditDistance {
        public double deleteWeight(char c) {
            return (Character.isLetter(c) || Character.isDigit(c)) ? -1 : 0;
        }
        public double insertWeight(char c) {
            return deleteWeight(c);
        }
        public double substituteWeight(char cDeleted, char cInserted) {
            return (Character.toLowerCase(cDeleted) == Character
                    .toLowerCase(cInserted)) ? 0 : -1;
        }
        public double matchWeight(char cMatched) {
            return 0;
        }
        public double transposeWeight(char cFirst, char cSecond) {
            return Double.NEGATIVE_INFINITY;
        }
    }
```

Figure C-10. String Comparison

When one-to-many matches are found during the AOLT selection process, the semantic surrounding of the source label is compared to the semantic surroundings of the matchees' semantic surrounding (the matchee can be a target label or a synonym of a target label) and the corresponding target label with the most similar semantic surrounding is chosen as the AOLT in SOCOM. Figure C-11 presents the ranking of semantic surroundings based on the edit distance concluded for a collection of character strings.

```java
    public String rankSurrounding(String[] sourceSurrounding, Object[] allConcernedTargetSurroundings) {
        String highestRanked = null;
        double stringScore = 0.0;
        double overallScore = 0.0;
        double avg = 0.0;
        double[] scores = new double[allConcernedTargetSurroundings.length];
        for (int i = 0; i < allConcernedTargetSurroundings.length; i++) {
            String[] eachConcernedTargetSurrounding = (String[]) allConcernedTargetSurroundings[i];
            overallScore = 0.0;
            for (int j = 0; j < eachConcernedTargetSurrounding.length; j++) {
                String s2 = eachConcernedTargetSurrounding[j];
                for (int k = 0; k < sourceSurrounding.length; k++){
                    String s1 = sourceSurrounding[k];
                    stringScore = getScore(s1, s2);
                    overallScore = overallScore + stringScore;
                    avg = overallScore / eachConcernedTargetSurrounding.length;
                }
            }
            scores[i] = avg;
        }
        double lowestScore = scores[0];
        String[] wantedObject = (String[])allConcernedTargetSurroundings[0];
        highestRanked = wantedObject[0];
        for(int i=1; i<scores.length; i++){
            if (scores[i] < lowestScore {
                lowestScore = scores[i];
                wantedObject = (String[])allConcernedTargetSurroundings[i];
                highestRanked = wantedObject[0];
            }
        }
        return highestRanked;
    }
```

Figure C-11. Semantic Surrounding Comparison

The code snippet in figure C-12 shows how translation collisions are resolved in SOCOM.

```java
    public String[] resolveCollision(String existingSourceTerm, String howExistingDerived, String newSourceTerm, String howNewDerived, String
        String[] resolvedResults = new String[2];
        if (howExistingDerived.equals("derivedFromTarget") && howNewDerived.equals("derivedWithoutTarget")){
            final String translationRepositoryURI = "C:\\Users\\BoFu\\Documents\\SOMMO\\Translations\\HTswrc_v0.3TranslationRepository.xml";
            SAXBuilder builder = new SAXBuilder();
            Document candidateDoc = builder.build(translationRepositoryURI);
            XPath path = XPath.newInstance("/TranslationResults/Result[SourceValue=\"" + newSourceTerm + "\"]");
            Element resultNode = (Element)path.selectSingleNode(candidateDoc);
            if (resultNode != null){
                resolvedResults[0] = collidedTranslation;
                Element candidateCollectionNode = resultNode.getChild("CandidateCollection");
                List candidates = candidateCollectionNode.getChildren("Candidate");
                for (int j = 0; j < candidates.size(); j++){
                    Element candidateValueNode = ((Element) candidates.get(j)).getChild("CandidateValue");
                    String candidateValueContent = candidateValueNode.getText();
                    if (!candidateValueContent.equals(collidedTranslation)){
                        resolvedResults[1] = candidateValueContent;
                    }
                }
            }
        }else if (howExistingDerived.equals("derivedWithoutTarget") && howNewDerived.equals("derivedFromTarget")){
            final String translationRepositoryURI = "C:\\Users\\BoFu\\Documents\\SOMMO\\Translations\\HTswrc_v0.3TranslationRepository.xml";
            SAXBuilder builder = new SAXBuilder();
            Document candidateDoc = builder.build(translationRepositoryURI);
            XPath path = XPath.newInstance("/TranslationResults/Result[SourceValue=\"" + existingSourceTerm + "\"]");
            Element resultNode = (Element)path.selectSingleNode(candidateDoc);
            if (resultNode != null){
                resolvedResults[1] = collidedTranslation;
                Element candidateCollectionNode = resultNode.getChild("CandidateCollection");
                List candidates = candidateCollectionNode.getChildren("Candidate");
                for (int j = 0; j < candidates.size(); j++){
                    Element candidateValueNode = ((Element) candidates.get(j)).getChild("CandidateValue");
                    String candidateValueContent = candidateValueNode.getText();
                    if (!candidateValueContent.equals(collidedTranslation)){
                        resolvedResults[0] = candidateValueContent;
                    }
                }
            }
        }else if (howExistingDerived.equals("derivedFromTarget") && howNewDerived.equals("derivedFromTarget")){
            final String translationRepositoryURI = "C:\\Users\\BoFu\\Documents\\SOMMO\\Translations\\HTswrc_v0.3TranslationRepository.xml";
            SAXBuilder builder = new SAXBuilder();
            Document candidateDoc = builder.build(translationRepositoryURI);
            XPath path = XPath.newInstance("/TranslationResults/Result[SourceValue=\"" + newSourceTerm + "\"]");
            Element resultNode = (Element)path.selectSingleNode(candidateDoc);
            if (resultNode != null){
                resolvedResults[0] = collidedTranslation;
                Element candidateCollectionNode = resultNode.getChild("CandidateCollection");
                List candidates = candidateCollectionNode.getChildren("Candidate");
                for (int j = 0; j < candidates.size(); j++){
                    Element candidateValueNode = ((Element) candidates.get(j)).getChild("CandidateValue");
                    String candidateValueContent = candidateValueNode.getText();
                    if (!candidateValueContent.equals(collidedTranslation)){
                        resolvedResults[1] = candidateValueContent;
                    }
                }
            }
        }else if (howExistingDerived.equals("derivedWithoutTarget") && howNewDerived.equals("derivedWithoutTarget")){
            final String translationRepositoryURI = "C:\\Users\\BoFu\\Documents\\SOMMO\\Translations\\HTswrc_v0.3TranslationRepository.xml";
            SAXBuilder builder = new SAXBuilder();
            Document candidateDoc = builder.build(translationRepositoryURI);
            XPath path = XPath.newInstance("/TranslationResults/Result[SourceValue=\"" + newSourceTerm + "\"]");
            Element resultNode = (Element)path.selectSingleNode(candidateDoc);
            if (resultNode != null) {
                resolvedResults[0] = collidedTranslation;
                Element candidateCollectionNode = resultNode.getChild("CandidateCollection");
                List candidates = candidateCollectionNode.getChildren("Candidate");
                for (int j = 0; j < candidates.size(); j++){
                    Element candidateValueNode = ((Element) candidates.get(j)).getChild("CandidateValue");
                    String candidateValueContent = candidateValueNode.getText();
                    if (!candidateValueContent.equals(collidedTranslation)){
                        resolvedResults[1] = candidateValueContent;
                    }
                }
            }
        }
    return resolvedResults;
```

Figure C-12. Collision Resolution in SOCOM

C.4. Prototype Two: SOCOM++ Code Snippets

This section presents some of the code used to implement the second prototype: SOCOM++.

Figure C-13 demonstrates how synonyms in English are generated for a candidate translation of a source label via the Big Huge Thesaurus API.

```java
String apiKey = "5e15fe8d66dcf679a342561f143ea510";
String format = "xml";
if(candidateTranslation != null){
    setProxy();
    URL url = new URL("http://words.bighugelabs.com/api/2/" + apiKey + "/" + candidateTranslation + "/" + format);
    HttpURLConnection c = (HttpURLConnection) url.openConnection();
    if(c.getResponseCode() == 200 || c.getResponseCode() == 303 ){
        BufferedReader br = new BufferedReader(new InputStreamReader(c.getInputStream()));
        StringBuffer sb = new StringBuffer();
        String line;
        while ((line = br.readLine()) != null){
            sb.append(line);
        }
        br.close();
        String result = sb.toString();
        DocumentBuilderFactory factory =  DocumentBuilderFactory.newInstance();
        InputSource source = new InputSource(new StringReader(result));
        Document doc = factory.newDocumentBuilder().parse(source);
        NodeList wList = doc.getElementsByTagName("w");
        for(int i=0; i<wList.getLength(); i++){
            Element wNode = (Element) wList.item(i);
            String attR = wNode.getAttribute("r");
            if(attR.equals("syn")){
                String synText = wNode.getTextContent();
                String[] synDetail = new String[3];
                synDetail[1] = synText;
                synDetail[0] = Integer.toString(counter);
                synDetail[2] = "BHT";
                candidateSynonyms.add(synDetail);
            }
            counter++;
        }
    }
}
Object[] allDetail = new Object[candidateSynonyms.size()];
for(int i = 0; i<candidateSynonyms.size(); i++){
    allDetail[i] = candidateSynonyms.get(i);
}
```

Figure C-13. English Synonym Generation via the Big Huge Thesaurus API

Figure C-14 demonstrates how synonyms in French are generated for target resource labels from the synonyms-fr.com.

```java
if(label != null){
    setProxy();
    String pageURL = "http://www.synonyms-fr.com/synonym.php?word=" + label + "&submit.x=0&submit.y=0";
    URL url = new URL(pageURL);
    URLConnection connection = url.openConnection();
    connection.setRequestProperty("Accept-Charset", "UTF-8");
    HasAttributeFilter filter = new HasAttributeFilter("onClick");
    org.htmlparser.Parser parser = new org.htmlparser.Parser(connection);
    org.htmlparser.util.NodeList list = parser.parse(filter);
    if (list.size() != 0){
        synonyms = new Object[list.size()];
        for (int i = 0; i < list.size(); i++){
            Node n = list.elementAt(i);
            String syn = n.getFirstChild().getText();
            String[] each = new String[3];
            each[1] = syn.trim();
            each[0] = Integer.toString(counter);
            each[2] = "synonyms-fr.com";
            synonyms[i] = each;
            counter++;
        }
    }
}
```

Figure C-14. French Synonym Generation via synonyms-fr.com

The code snippet shown in figure C-15 illustrates how property validation is achieved in SOCOM++. More specifically, a message (either `success` or `error`) is returned upon the completion of validation. In the case of an error, instructions are presented to the user for the correction configurations of the `properties.xml` file in the system.

```java
        if(taskIntentCorrectness = (true || false)){
            message = "success";
        }else{
            message = "error";
            System.out.println("The value of the <entry key=\"correctnessOptimise\"/> element must be set to either true or false.");
        }
        if(taskIntentCompleteness = (true || false)){
            message = "success";
        }else{
            message = "error";
            System.out.println("The value of the <entry key=\"completenessOptimise\"/> element must be set to either true or false.");
        }
        if(pseudoFeedback > 0.0 && pseudoFeedback < 1.0){
            message = "success";
        }else{
            message = "error";
            System.out.println("The value of the <entry key=\"threshold\"/> element must be set to a value between zero and one.");
        }
        if(timeConstraint==false &&
                ((timeConstraint||sourceSemantics||targetSemantics||resourceConstraintSource||
                 resourceConstraintTarget||taskIntentCorrectness||taskIntentCompleteness)==true ||
                  (pseudoFeedback>0.0&&pseudoFeedback<1.0))){
            message = "success";
        }else{
            message = "error";
            System.out.println("When the value of the <entry key=\"default\"/> element is set to false, " +
                    "at least one of the other elements' value should be set.");
        }
        if(timeConstraint==false && ((taskIntentCorrectness==true && taskIntentCompleteness==false) ||
                (taskIntentCorrectness==false && taskIntentCompleteness==true))){
            message = "success";
        }else{
            message = "error";
            System.out.println("Only one of the <entry key=\"correctnessOptimise\"/> and " +
                    "the <entry key=\"completenessOptimise\"/> element can be set to true at a time.");
        }
```

Figure C-15. Property Validation

Figure C-16 presents a code snippet of example use of XQuery via the XML:DB API. It illustrates how a candidate AOLT with specified attribute values (i.e. with a precise `sourceID`, a particular `translationSource` and a specified `type` in the AOLT record) is retrieved from the database in SOCOM++.

```java
    public Object[] lookupAOLTRecord(String sourceID, String media, String type, XMLResource aoltRecord) throws XMLDBException{
        Object[] aolt = new Object[2];
        Collection c = aoltRecord.getParentCollection();
        XQueryService s = (XQueryService) c.getService("XQueryService", "1.0");
        s.setProperty("indent", "yes");
        String arg = "for $x in doc(\"" + aoltRecord.getDocumentId() + "\")//Record[@sourceID=\""
                    + sourceID + "\" and @translationSource=\"" + media + "\" and @type=\"" + type + "\"] return $x";
        CompiledExpression expression = s.compile(arg);
        ResourceSet set = s.execute(expression);
        if(set!=null && set.getSize()>0){
            XMLResource recordRes = (XMLResource) set.getResource(0);
            Document recordDOM = (Document) recordRes.getContentAsDOM();
            Element recordEle = (Element) recordDOM.getElementsByTagName("Record").item(0);
            String sourceValue = recordEle.getAttribute("sourceValue");
            String translation = recordEle.getAttribute("aoltValue");
            aolt[0] = true;
            String[] info = new String[5];
            info[0] = sourceID;
            info[1] = sourceValue;
            info[2] = translation;
            info[3] = media;
            info[4] = type;
            aolt[1] = info;
        }else{
            aolt[0] = false;
        }
        return aolt;
    }
```

Figure C-16. XQuery Example

Figure C-17 presents a code snippet of an example use of XPathQuery via the XML:DB API. It illustrates how a set of `<Record/>` elements with specified attribute values (i.e. with a given `sourceID` and is of a particular `type`) in the AOLT record is retrieved in SOCOM++.

```
        Collection col = aoltRecord.getParentCollection();
        XPathQueryService service = (XPathQueryService) col.getService("XPathQueryService", "1.0");
        service.setProperty("indent", "yes");
        ResourceSet typeOneSet = service.queryResource(aoltRecord.getDocumentId(),
                "//Record[@sourceID=\"" + sourceID + "\" and @type=\"one\"]");
        ResourceSet typeTwoSet = service.queryResource(aoltRecord.getDocumentId(),
                "//Record[@sourceID=\"" + sourceID + "\" and @type=\"two\"]");
        ResourceSet typeThreeSet = service.queryResource(aoltRecord.getDocumentId(),
                "//Record[@sourceID=\"" + sourceID + "\" and @type=\"three\"]");
        ResourceSet typeFourSet = service.queryResource(aoltRecord.getDocumentId(),
                "//Record[@sourceID=\"" + sourceID + "\" and @type=\"four\"]");
        ResourceSet typeFiveSet = service.queryResource(aoltRecord.getDocumentId(),
                "//Record[@sourceID=\"" + sourceID + "\" and @type=\"five\"]");
        ResourceSet typeSixSet = service.queryResource(aoltRecord.getDocumentId(),
                "//Record[@sourceID=\"" + sourceID + "\" and @type=\"six\"]");
```

Figure C-17. XPathQuery Example

APPENDIX D. DOCUMENT TYPE DEFINITIONS

D.1. Appendix Overview

This Appendix contains the Document Type Definitions (DTDs) for the XML documents created during this Ph.D. Section D.2 presents the DTDs used by prototype one: SOCOM. Section D.3 presents the DTDs declared by the second prototype: SOCOM++.

D.2. DTDs in Prototype One: SOCOM

This section contains the DTDs used by the first prototype. Figure D-1 below illustrates the DTD for the translation repository.

```
<!ELEMENT TranslationRepository (Result*)>
<!ELEMENT Result (SourceID, SourceValue, CandidateCollection)>
<!ELEMENT SourceID (#PCDATA)>
<!ELEMENT SourceValue (#PCDATA)>
<!ELEMENT CandidateCollection (Candidate*)>
<!ELEMENT Candidate (CandidateID, CandidateValue)>
<!ELEMENT CandidateID (#PCDATA)>
<!ELEMENT CandidateValue (#PCDATA)>
```

Figure D-1. Translation Repository DTD

Figure D-2 presents the DTD for the Lexicon repository developed in SOCOM.

```
<!ELEMENT LexiconRepository (Result*)>
<!ELEMENT Result (TargetID, TargetValue, SynonymCollection)>
<!ELEMENT TargetID (#PCDATA)>
<!ELEMENT TargetValue (#PCDATA)>
<!ELEMENT SynonymCollection (Synonym*)>
<!ELEMENT Synonym (SynonymID, SynonymValue)>
<!ELEMENT SynonymID (#PCDATA)>
<!ELEMENT SynonymValue (#PCDATA)>
```

Figure D-2. Lexicon Repository DTD

D.3. DTDs in Prototype Two: SOCOM++

This section contains the DTDs used by the second prototype: SOCOM++. Figure D-3 below illustrates the DTD for the configuration file.

```
<!ELEMENT properties ( comment?, entry* ) >
<!ATTLIST properties version CDATA #FIXED "1.0">
<!ELEMENT comment (#PCDATA) >
<!ELEMENT entry (#PCDATA) >
<!ATTLIST entry key CDATA #REQUIRED>
```

Figure D-3. Configuration File DTD[108]

[108] http://java.sun.com/dtd/properties.dtd

Figure D-4 contains the DTD declared for storing the source semantics analysed.

```
<!ELEMENT SourceSemantic (Resource*)>
<!ATTLIST Resource id ID #REQUIRED>
<!ELEMENT Resource (OntLabel, MTLabel, Translation*, Surrounding*)>
<!ELEMENT OntLabel (#PCDATA)>
<!ELEMENT MTLabel (#PCDATA)>
<!ELEMENT Translation (Candidate*)>
<!ELEMENT Surrounding EMPTY>
<!ATTLIST Surrounding id IDREF #REQUIRED>
<!ATTLIST Surrounding OntLabel CDATA #IMPLIED>
<!ATTLIST Surrounding MTLabel CDATA #IMPLIED>
<!ELEMENT Candidate (CandidateValue, CandidateSource,
CandidateConcatenated, CandidateSynonymCollection?)>
<!ATTLIST Candidate id ID #REQUIRED>
<!ELEMENT CandidateValue (#PCDATA)>
<!ELEMENT CandidateSource (#PCDATA)>
<!ELEMENT CandidateConcatenated (#PCDATA)>
<!ELEMENT CandidateSynonymCollection (CandidateSynonym*)>
<!ELEMENT CandidateSynonym EMPTY>
<!ATTLIST CandidateSynonym id ID #REQUIRED>
<!ATTLIST CandidateSynonym source CDATA #IMPLIED>
<!ATTLIST CandidateSynonym value CDATA #IMPLIED>
<!ATTLIST CandidateSynonym concatenated CDATA #IMPLIED>
```

Figure D-4. Source Semantic DTD

Figure D-5 presents the DTD declared for storing the analysed target semantics.

```
<!ELEMENT TargetSemantic (Resource*)>
<!ATTLIST Resource id ID #REQUIRED>
<!ELEMENT Resource (OntLabel, MTLabel, SynonymCollection?,
Surrounding*)>
<!ELEMENT OntLabel (#PCDATA)>
<!ELEMENT MTLabel (#PCDATA)>
<!ELEMENT SynonymCollection (Synonym*)>
<!ELEMENT Synonym EMPTY>
<!ATTLIST Synonym id ID #REQUIRED>
<!ATTLIST Synonym source CDATA #IMPLIED>
<!ATTLIST Synonym value CDATA #IMPLIED>
<!ATTLIST Synonym concatenated CDATA #IMPLIED>
<!ELEMENT Surrounding EMPTY>
<!ATTLIST Surrounding id IDREF #REQUIRED>
<!ATTLIST Surrounding OntLabel CDATA #IMPLIED>
<!ATTLIST Surrounding MTLabel CDATA #IMPLIED>
```

Figure D-5. Target Semantic DTD

Figure D-6 presents the DTD declared for the AOLT record in SOCOM++.

```
<!ELEMENT AOLTRecord (Record*)>
<!ELEMENT Record EMPTY>
<!ATTLIST Record sourceID CDATA #REQUIRED>
<!ATTLIST Record sourceValue CDATA #IMPLIED>
<!ATTLIST Record translationSource CDATA #IMPLIED>
<!ATTLIST Record type CDATA #IMPLIED>
<!ATTLIST Record aoltID CDATA #REQUIRED>
<!ATTLIST Record aoltValue CDATA #IMPLIED>
```

Figure D-6. AOLT Record DTD

Figure D-7 presents the DTD declared for the AOLT selection in SOCOM++.

```
<!ELEMENT AOLTSelection (AOLT*)>
```

```
<!ELEMENT AOLT EMPTY>
<!ATTLIST AOLT sourceID ID #REQUIRED>
<!ATTLIST AOLT media CDATA #IMPLIED>
<!ATTLIST AOLT type CDATA #IMPLIED>
<!ATTLIST AOLT source CDATA #IMPLIED>
<!ATTLIST AOLT translation CDATA #REQUIRED>
```

Figure D-7. AOLT Selection DTD

Figure D-8 presents the DTD declared for the output generated from task intent analysis in SOCOM++.

```
<!ELEMENT TaskIntent (Entry*)>
<!ELEMENT Entry EMPTY>
<!ATTLIST TaskIntent algorithm CDATA #REQUIRED>
<!ATTLIST TaskIntent intent CDATA #REQUIRED>
<!ATTLIST TaskIntent matches CDATA #IMPLIED>
<!ATTLIST TaskIntent estimate CDATA #IMPLIED>
<!ATTLIST Entry count CDATA #REQUIRED>
<!ATTLIST Entry media CDATA #REQUIRED>
<!ATTLIST Entry type CDATA #REQUIRED>
<!ATTLIST Entry usage CDATA #REQUIRED>
```

Figure D-8. Task Intent DTD

Figure D-9 presents the DTD declared for the output generated from pseudo feedback in SOCOM++.

```
<!ELEMENT PseudoFeedback (Entry*)>
<!ELEMENT Entry EMPTY>
<!ATTLIST PseudoFeedback algorithm CDATA #REQUIRED>
<!ATTLIST PseudoFeedback threshold CDATA #REQUIRED>
<!ATTLIST PseudoFeedback matches CDATA #IMPLIED>
<!ATTLIST PseudoFeedback estimate CDATA #IMPLIED>
<!ATTLIST Entry count CDATA #REQUIRED>
<!ATTLIST Entry media CDATA #REQUIRED>
<!ATTLIST Entry type CDATA #REQUIRED>
<!ATTLIST Entry usage CDATA #REQUIRED>
```

Figure D-9. Pseudo Feedback DTD

APPENDIX E. CONFIDENCE LEVEL EVALUATION PLOTS

This appendix contains the plots generated during the evaluation of SOCOM++. The legends used in these plots are as follows:

▲ Baseline ▲ SOCOM++ ● Baseline Mean ● SOCOM++ Mean

E.1. Trial One

Figure E-1 shows the plots generated for the confidence level evaluation in trial one. As shown in both plots, the green triangles (illustrating the results generated by SOCOM++) are mostly located on the top left corner of the plot, comparing to the orange triangles – representing the results generated by the baseline system. This suggests that the matches generated by SOCOM++ are more confident and less dispersed than those generated by the baseline system in both experiments.

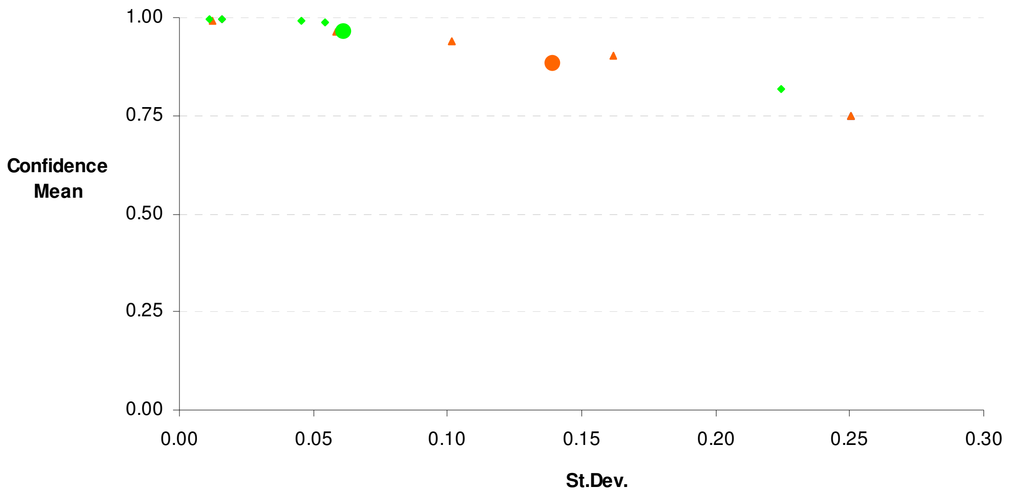
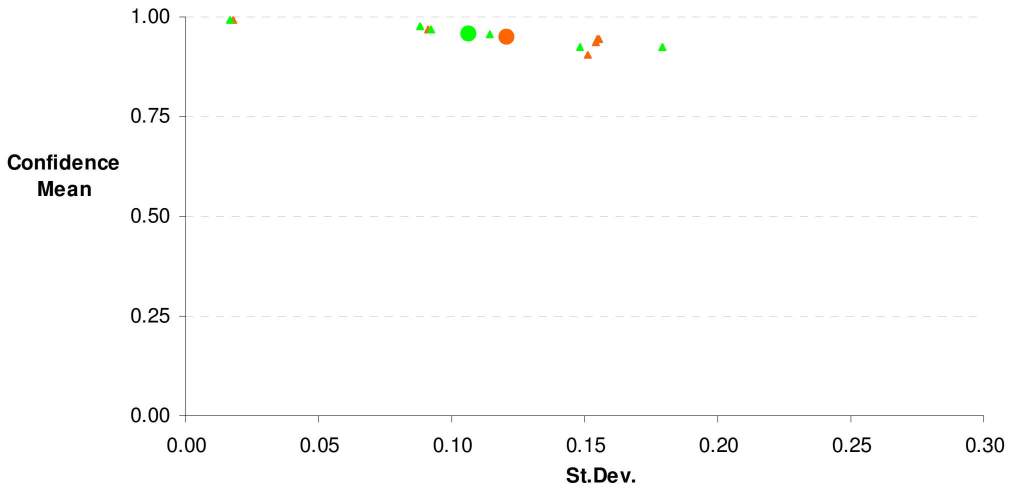
Exp.	Confidence Mean & St. Dev.
1	
2	

Figure E-1. Confidence Evaluation in Trial One

E.2. Trial Two

Figure E-2 presents the plots generated during the confidence level evaluation in trial two. In experiment one, green dots are mostly located on the top left corner of the graph, whereas the orange dots are mostly located at the bottom right corner. This means that the matches generated by SOCOM++ trial two are more confident and less dispersed than those generated by the baseline system in both experiments. This is not the case for experiment two however, improvement is not shown in either the mean of the confidence levels nor their standard deviations.

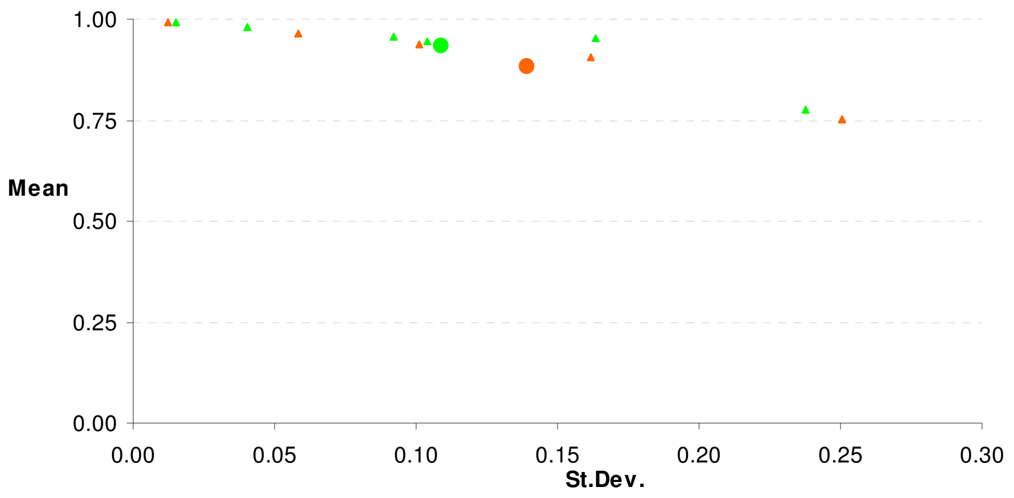
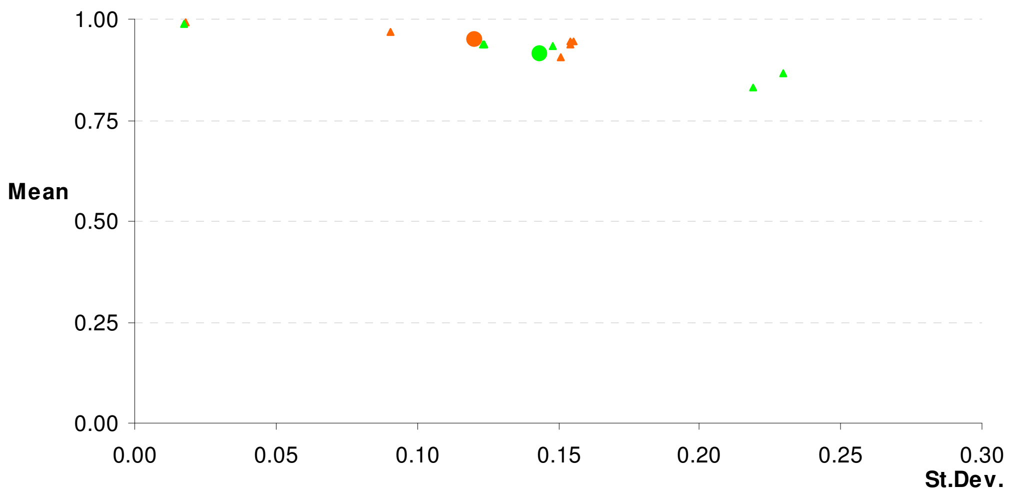
Exp.	Confidence Mean & St. Dev.
1	
2	

Figure E-2. Confidence Evaluation in Trial Two

E.3. Trial Three

Figure E-3 shows the scatter plots of the confidence level evaluation in trial three. In both experiments, matches were found to be less confident and with more dispersed confidence levels (i.e. green dots are mostly scattered around the bottom right corner of the graph) when applying the SOCOM++ trial three configuration.

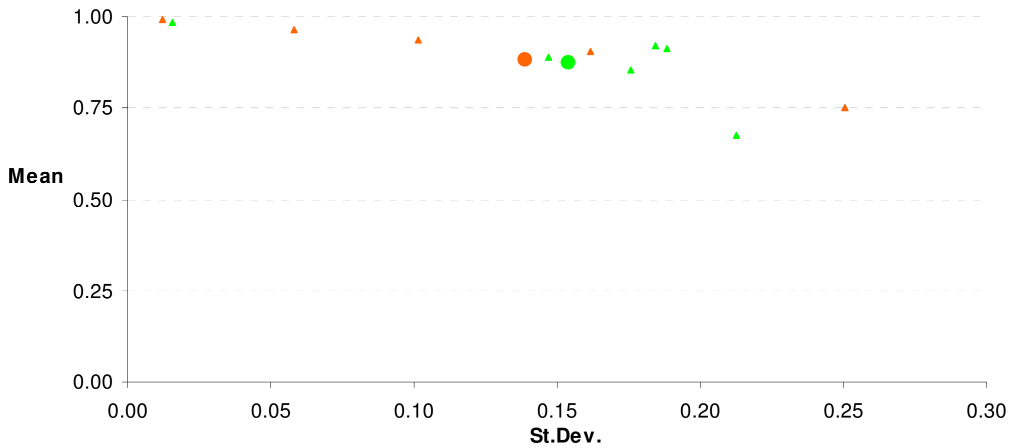
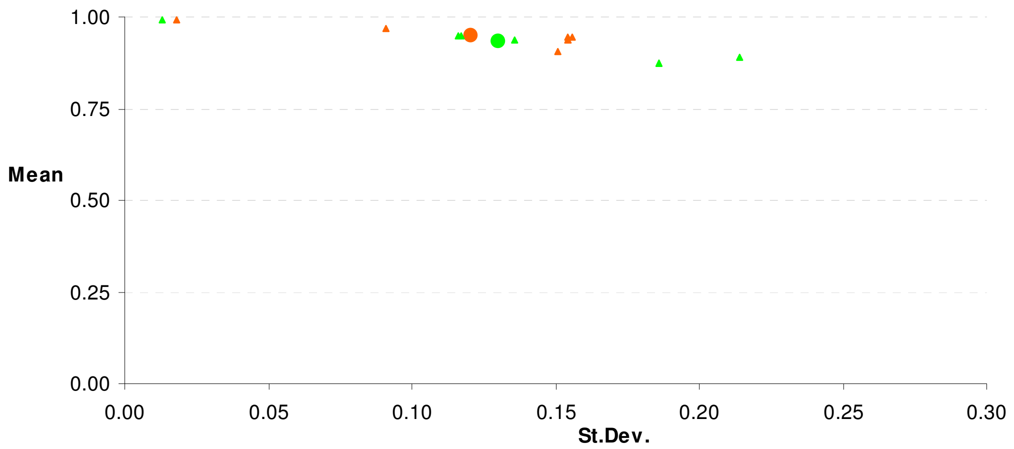
Exp.	Confidence Mean & St. Dev.
1	
2	

Figure E-3. Confidence Evaluation in Trial Three

E.4. Trial Four

Figure E-4 presents the scatter plots generated for the confidence level evaluation in trial four. Improvement (i.e. higher mean confidence level and lower standard deviation) when mapping the CSWRC ontology (in Chinese) to the ISWC ontology (in English) is more evident (i.e. green dots are mostly located at the top left corner of the graph) than when mapping the 101 ontology (in English) to the 206 ontology (in French).

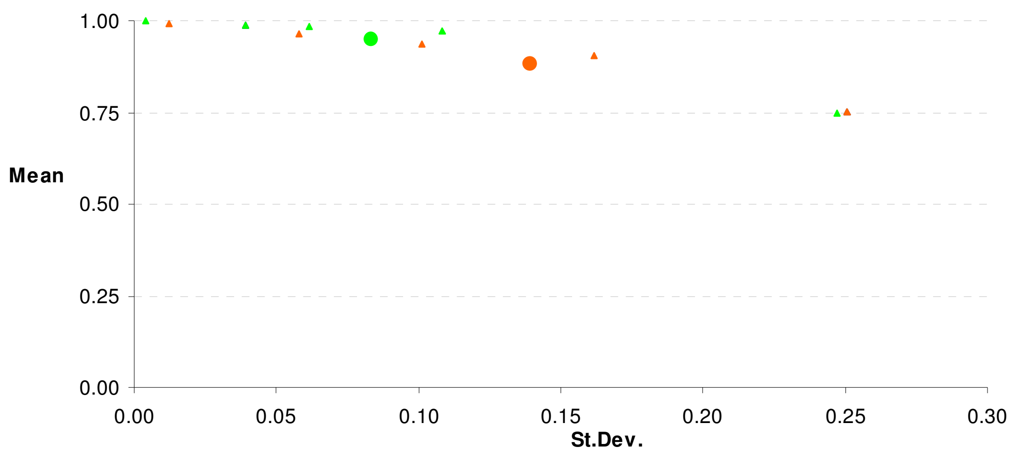
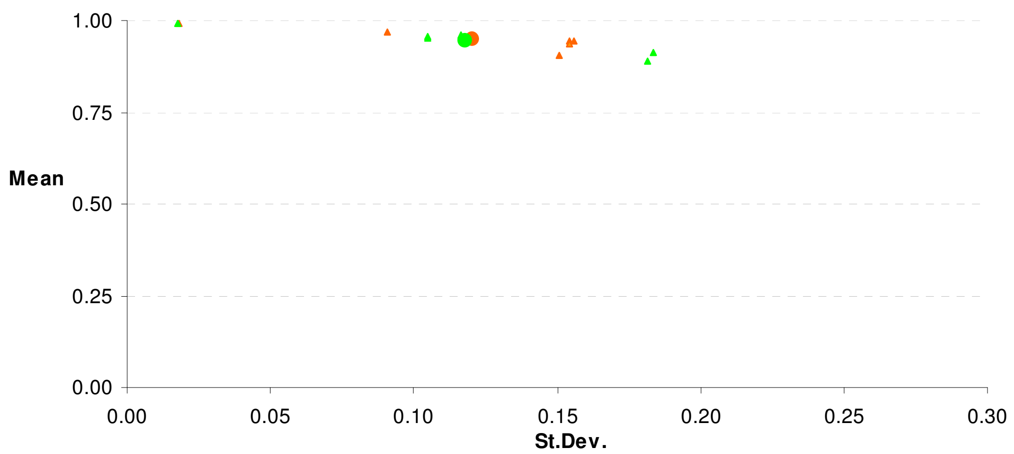
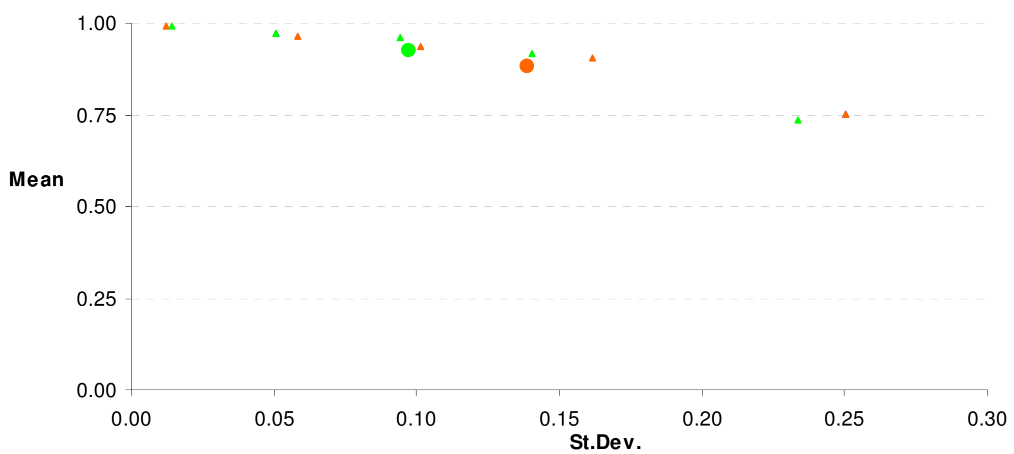
Exp.	Confidence Mean & St. Dev.
1	
2	

Figure E-4. Confidence Evaluation in Trial Four

E.5. Trial Five

Figure E-5 presents the scatter plots generated in trial five. Greater improvement on the confidence levels (i.e. higher mean confidence level and lower standard deviation) can be seen in experiment one - mapping the CSWRC ontology to the ISWC ontology (i.e. green dots are mostly located to the top left corner of the graph).

Exp.	Confidence Mean & St. Dev.
1	

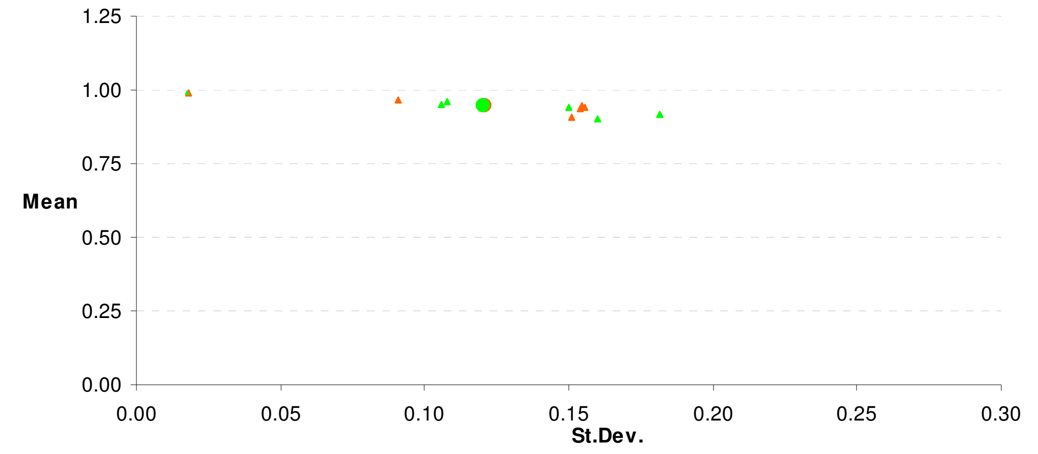
<table>
<tr><td>2</td><td>
</td></tr>
</table>

Figure E-5. Confidence Evaluation in Trial Five

E.6. Trial Six

Figure E-6 presents the scatter plots generated for the confidence level evaluation in trial six. Improvements on mean and standard deviation in SOCOM++ trial six can be see in experiment one. In experiment two however, though higher confidence mean was found in SOCOM++, more dispersed confidence levels were also evident.

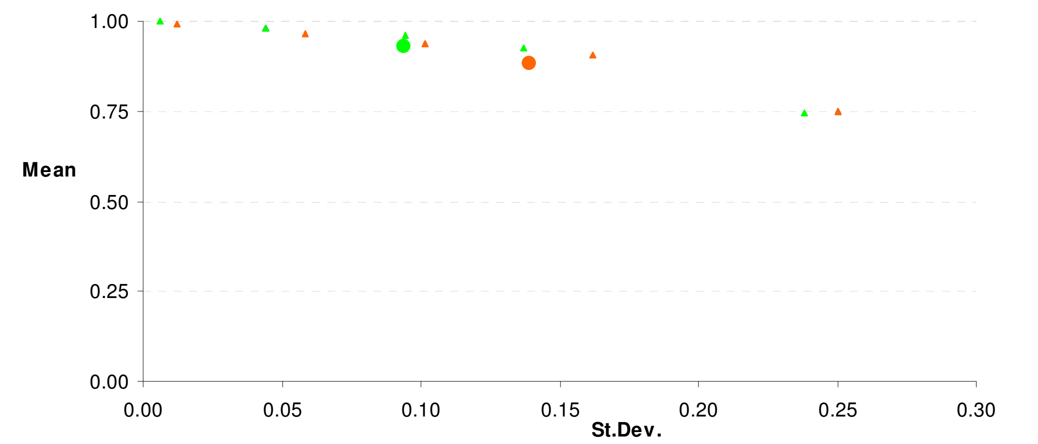
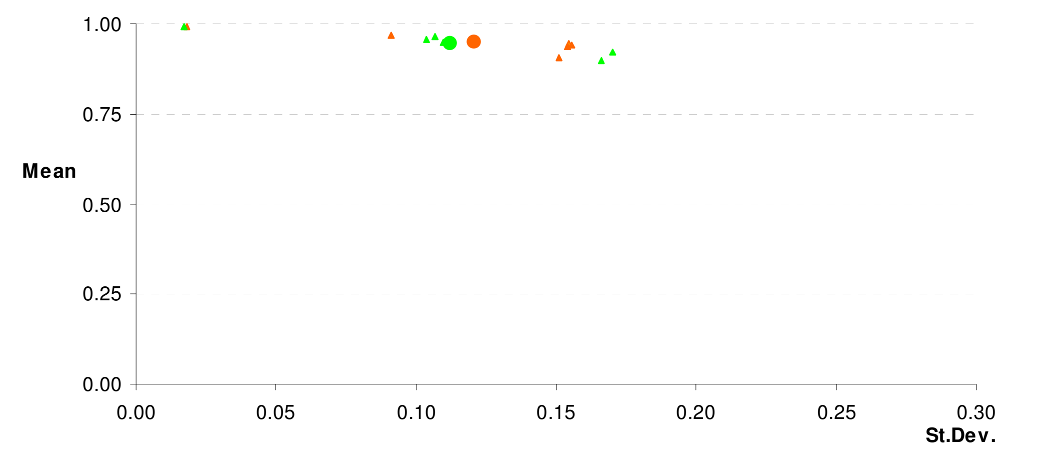
<table>
<tr><th>Exp.</th><th>Confidence Mean & St. Dev.</th></tr>
<tr><td>1</td><td>
</td></tr>
<tr><td>2</td><td>
</td></tr>
</table>

Figure E-6. Confidence Evaluation in Trial Six