A Configurable Translation-Based Cross-Lingual Ontology Mapping System to adjust Mapping Outcome

Bo Fu*, Rob Brennan, Declan O’Sullivan
Knowledge and Data Engineering Group, School of Computer Science and Statistics,
Trinity College Dublin, Ireland
{bofu, rob.brennan, declan.osullivan}@scss.tcd.ie

Abstract

Ontologies are widely considered as the building blocks of the semantic web, and with them, comes the data interoperability issue. As ontologies are not necessarily always labelled in the same natural language, one way to achieve semantic interoperability is by means of cross-lingual ontology mapping. Translation techniques are often used as an intermediate step to translate the conceptual labels within an ontology. This approach essentially removes the natural language barrier in the mapping environment and enables the application of monolingual ontology mapping tools. This paper shows that the key to this translation-based approach to cross-lingual ontology mapping lies with selecting appropriate ontology label translations in a given mapping context. Appropriateness of the translations in the context of cross-lingual ontology mapping differs from the ontology localisation point of view, as the former aims to generate correct mappings whereas the latter aims to adapt specifications of conceptualisations to target communities. This paper further demonstrates that the mapping outcome using the translation-based cross-lingual ontology mapping approach is conditioned on the translations selected for the intermediate label translation step. In particular, this paper presents the design, implementation and evaluation of a novel cross-lingual ontology mapping system: SOCOM++. SOCOM++ provides configurable properties that can be manipulated by a user in the process of selecting label translations in an effort to adjust the subsequent mapping outcome. It is shown through the evaluation that for the same pair of ontologies, the mappings between them can be adjusted by tuning the translations for the ontology labels. This finding is not yet shown in previous research.

Keywords: Cross-Lingual Ontology Mapping; Semantic-Oriented Configurable Ontology Label Translation; Adjustable Mapping Outcome.

1. Introduction

Ontologies, as specifications of conceptualisations [19], are recognised as a “basic component of the semantic web” [4] and have been widely used in knowledge management [26]. One approach to ontology construction is to use language-neutral identifiers to label concepts [36], whereby ontological resources are natural language independent. However, the benefits of this approach are debatable, as Bateman points out “the path towards viable ontologies is one that is irreconcilably connected to natural language” [2]. In practice, natural language labels are commonly used in ontological resource naming as seen in [38, 21]. As a result, ontologies that are labelled in diverse natural languages are increasingly evident (discussed in section 2.2). Given ontologies that are likely to be authored by different actors using different terminologies, structures and natural languages, ontology mapping 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. Little research has focused on mapping scenarios where the ontologies involved are labelled in different natural languages. However, current monolingual mapping techniques often rely on lexical comparisons made between resource labels, which limits their deployment to ontologies in the same natural language or at least in comparable natural languages[1]. 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), a match to the class `<owl:Class rdf:about="#奶酪">` in the target ontology (meaning cheese in Chinese) may be neglected. Even though multilingual support can be provided to ontologies via language tagging, this form of assistance may not always be available to every mapping scenario. For example, in Fig. 1[2], *CoberturaDeQueijo* tags the label of the `CheeseTopping` class in Portuguese. Assuming the ontology to be mapped to is also in Portuguese, the content in `rdfs:label` may then be used by monolingual matching tools. However, such an approach requires all the resources in a given ontology to be tagged with target natural language content, which may be challenging since this is not a requirement in formal ontologies.

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

Fig. 1. An example of associating multilingual natural language content to resources using rdfs:label. The *CheeseTopping* class is defined as a subclass of *PizzaTopping*, and is also tagged with `CoberturaDeQueijo` in Portuguese.

[*] Corresponding author. Present address: bofu@uvic.ca, Computer Human Interaction and Software Engineering Lab, Department of Computer Science, University of Victoria, British Columbia, Canada

[1] An example of comparable natural languages can be English and French, or Italian and German - regardless of the language family they belong to, they are all alphabetic letter-based with 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 are logogram-based and the graphemes in the latter are alphabetic letter-based. Note that in this context, comparable natural languages are not necessarily from the same language family.

[2] This snippet is taken from the Pizza ontology at http://smi-protege.stanford.edu/svn/owl/trunk/examples/pizza.owl?rev=10355&view=auto

Given the limitations of existing monolingual mapping tools, there is a pressing need for the development of matching techniques that can work with ontologies in different natural languages. One approach is cross-lingual ontology mapping. In this paper, *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 [57, 5, 54, 51] 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 mapping (MOM) tools. Such an approach is referred to as the translation-based cross-lingual ontology mapping approach in this paper. The typical process involved in a translation-based CLOM approach 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 using MOM techniques. The intermediate step concerning the translation of ontology labels is often achieved by using machine translation (MT) techniques. Various techniques [9] such as statistical MT and rule-based MT have been developed, which aim to improve the quality of translations through word sense disambiguation (WSD) [34]. In other words, MT tools are intended to assign an accurate meaning to a phrase in a specific natural language while limiting possible ambiguity, which is not necessarily a requirement in CLOM however. This is because to achieve CLOM, translations should lead to the generation of correct mappings, but it is not of interest whether these translations are the most accurate localisations in the specific natural language. Consequently, translating the ontology labels in the context of CLOM is not solely concerned with finding translated equivalents in the target natural language, but also finding translations that can lead to correct mappings. There can be various ways to express the same or similar concept in many natural languages. A simple example of this is: *Ph.D. candidate* and *doctoral student* both describe someone who is pursuing an academic degree of Doctor of Philosophy. Envision this in the context of CLOM, assuming the target ontology is labelled in English and the source ontology is labelled in a natural language other than English. For a concept in the source ontology, its English translation can be *Ph.D. candidate* or *doctoral student*. Which one 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 correct mapping given that an equivalent concept is also presented in the target ontology. This translation selection process differs from traditional word sense disambiguation, as WSD is “the association of a given word in a text or discourse with a definition or meaning (sense) which is distinguish-able from other meanings potentially attributable to that word” [23]. In the context of translation-based CLOM, the outcome of the mapping process relies on the translations selected for the given ontology labels as previously demonstrated in [17, 18].

The objective of this paper is to investigate how the translations of ontology labels can be adjusted in order to alter CLOM outcome, given that it is likely there being many ways to describe the same concept. In particular, this paper argues that ontology label translations should not take place in isolation from the ontologies involved in the mapping context. To facilitate translations that are conducted for the purpose of CLOM, this paper presents a configurable cross-lingual ontology mapping system: SOCOM++ (Semantic-Oriented Cross-Lingual Ontology Mapping), which is designed specifically for adjusting translations of ontology labels in an effort to improve the mapping outcome.

The key to SOCOM++ is that it consults the embedded semantics within the ontologies in a given CLOM scenario as well as background semantics when generating ontology label translations in the process of achieving CLOM. The meaning of *semantic* in the context of *semantic-oriented* cross-lingual ontology mapping is two-fold. On one hand, semantic refers to the specifications of conceptualisations in the ontology, i.e. the embedded semantic data coded in the ontology. In SOCOM++, this is illustrated by analysing the entity labels and their structural surroundings. On the other hand, semantic also refers to the meaning of the conceptualisations presented in the ontology. In SOCOM++, this is illustrated by the use of background semantics such as translations and synonyms of the entity labels that are available through external resources such as MT tools and thesauri.

The effectiveness of SOCOM++ is demonstrated through a set of reproducible experiments in this paper. The evaluation results show that the cross-lingual ontology mapping outcome is conditioned on the intermediate translations selected for the given ontology labels, and the mapping outcome between the same ontology pair is adjustable depending on the translations selected. The contribution of this research is twofold. First, it validates the importance of selecting appropriate ontology label translations (AOLT) in translation-based cross-lingual ontology mapping, where appropriateness is determined by whether a translation leads to a correct mapping. Second, it presents a mapping system: SOCOM++ that provides the necessary support for adjusting mapping outcomes through the alteration of entity label translations.

The focus of this research is formally defined ontologies that follow the Resource Description Framework [3] (RDF) schema or the Web Ontology Language[4] (OWL) specifications. The focus of the CLOM process shown in this paper is the generation of correspondences between ontological resources in formally defined multilingual ontologies. In this paper, *multilingual ontologies* refer to two or more 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 [45]). In addition, these ontologies have not been linguistically enriched (e.g. the ontological resources are associated with linguistic information as presented in [42]), nor do they associate multilingual natural language content for a given ontological resource (such as the

[3] http://www.w3.org/TR/rdf-schema

[4] http://www.w3.org/TR/owl-features

example shown in Fig. 1). Furthermore, *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[5]. 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.

The remainder of this paper is organised as follows. A state of the art review on CLOM approaches and related background are presented in section 2. In particular, the challenge of translations carried out for the purpose of CLOM is discussed and differentiated from translations carried out for the purpose of localisation. To tackle this challenge, SOCOM++ is proposed. Its design and implementation details are discussed in sections 3 and 4 respectively. An overview of the evaluation is presented in section 5. Three configurations of SOCOM++ (focusing on the adjustment of ontology semantics) and their findings are presented in section 6. Another three configurations (focusing on executing a second iteration of the AOLT selection process) and their findings are presented in section 7. A summary of the lessons learned is presented in section 8. Finally, section 9 concludes this paper with some suggestions for future work.

2. Background and State of the Art

This section presents related background and a review on current approaches used in cross-lingual ontology mapping.

2.1 The Ontology Mapping Problem

Ontologies have gained increasing interest for structured modelling of meaning from the semantic web community [32]. However, in decentralised environments such as the semantic web, the heterogeneity issue occurs when ontologies are created by different authors. 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. Ontologies being specifications of conceptualisations [19] are thus 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 instance. To address the heterogeneity issue arising from ontologies on the semantic web, ontology mapping has become an important research field [53].

In the literature, ontology matching (e.g. [15]), ontology mapping (e.g. [27]) and ontology alignment (e.g. [12]) are used interchangeably to refer to the process of correspondence generation between ontologies. Ontology matching and ontology mapping are differentiated by O'Sullivan et al. [40], 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 O'Sullivan et al.'s approach, in this paper, 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 paper. The SOCOM++ system presented in this paper aims to provide support to the cross-lingual ontology mapping process by generating candidate matches through the matching process.

The following definition for correspondence in ontology mapping is adopted in this paper: "*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: $\langle id, e, e', r, n\rangle$, 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 $\langle id, e, e', r, n\rangle$ asserts that the relation r holds between the ontology entities e and e' with confidence n.*" [14, p.46][6]

A set of alignment relations[7] "*correspond to set-theoretic relations between classes: equivalence (=); disjointness ($\perp$); more general ($\supseteq$) … 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*" [14, p.45]. A confidence structure is "*an ordered set of degrees $\langle \Xi, \leq \rangle$ for which there exists a greatest element $\top$ and a smallest element $\perp$*"[14, p.46]. In this paper, MOM results are generated using the Alignment API[8]. In this paper, equivalence relations *(=)* with confidence levels that range between 0.0 (i.e. the smallest element) and 1.0 (i.e. the greatest element) are generated. Equivalent correspondences are currently the dominant relations that are generated by MOM tools as evidently shown by the participating MOM systems in the ontology alignment evaluation initiative (OAEI) contests since 2004[9], thus are the focus of this research.

2.2 Ontologies and Multilinguality

Ontologies are widely used in knowledge-based systems and the applications of ontologies traverse many disciplines, discussed next. In agriculture, the Food and Agriculture Organization (FAO) provides reference standards for defining

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

[6] In this context, entity language refers to the ontology language, e.g. OWL, RDF, etc. In this paper, natural language refers to linguistic languages possessed by humans.

[7] The SKOS vocabulary defines mapping relations as `skos:exactMatch` - there is a high degree of confidence that the concepts can be used interchangeably; `skos:closeMatch` - a match between two concepts that are sufficiently similar and can be used interchangeably; `skos:relatedMatch` - an associative match between two concepts; `skos:broadMatch` and `skos:narrowMatch` - both state hierarchical matches between two concepts, see http://www.w3.org/2009/08/skos-reference/skos.html

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

[9] OAEI results since 2004 can be found at http://oaei.ontologymatching.org

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 [7], mapping methodologies for them [30, 31] and a definition of requirements to improve the interoperability between these multilingual resources [6]. In education, the Bologna declaration has introduced an ontology-based framework for qualification recognition [52] 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 spring 2012). In e-learning, educational ontologies are used to enhance learning experiences [10] and to empower system platforms with high adaptivity [46]. In finance, ontologies are used to model knowledge in the stock market domain [1] and portfolio management [56]. In medicine, ontologies are used to improve knowledge sharing and reuse, such as work presented by Fang et al. [16] that focuses on the creation of a traditional Chinese medicine ontology, and work presented by Tenenbaum et al. [49] that 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 authors, the heterogeneity issue is inevitable in the presence of ontologies that are built on different models of conceptualisations and natural languages. The very existence of ontologies in various natural languages provides an impetus to discover ways to support semantic interoperability.

Lexical databases, such as WordNet, can be considered as lightweight ontologies, as the terms in them often relate to one another via synonymic, antonymic etc. associations. According to the Global WordNet Association[10], at the time of this writing, there are more than forty lexicons in the world containing a collective set of over fifty different natural languages. These languages include Arabic (used in ArabicWordNet [11]); Bulgarian (used in BulNet[12]); Chinese (used in HowNet[13]); Dutch, French, German, Italian, Spanish (used in EuroWordNet[14]); Irish (used in LSG[15]) and many others. Multilinguality is also evident in formally defined ontologies. According to the OntoSelect Ontology Library[16], (in August 2011) more than 25% of the 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. [29] present a semi-automatic framework to generate multilingual ontologies in an attempt to reduce labour costs. Niwa et al. [37] define a formula to extract word relations based on document frequency and conditional probability. Srinivasan [47] conducted similar research and proposed an algorithm to generate hierarchies of words. Shimoji et al. [45] 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 activities highlight the support that is available for the creation of multilingual ontologies, and the need to achieve interoperability between them in the process of knowledge sharing and reuse. Current approaches that tackle cross-lingual ontology mapping are discussed next.

2.3 Cross-Lingual Ontology Mapping

Current approaches to CLOM can be grouped into five categories: manual CLOM [31], corpus-based CLOM [35], CLOM via linguistic enrichment [41], CLOM via indirect alignment [25] and translation-based CLOM [54, 50, 57]. 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 [31], where an English thesaurus: AGROVOC [17] (developed by the FAO containing a set of agricultural vocabularies) is mapped to a Chinese thesaurus: CAT [18] (Chinese Agricultural Ontology, developed by the Chinese Academy of Agricultural Science). The thesauri are loaded in the Protégé[19] editor, and segments of the thesauri are assigned to groups of terminologists to generate mappings. Finally, these manually generated mappings are reviewed and stored. The authors, 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 [35]. 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) is carried out to associate synsets[20] in the given thesauri. This approach is evaluated by a pair of domain 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 [28]), which makes the construction process easier. However, a disadvantage of using corpora is that the construction overhead could be a

[10] http://www.globalwordnet.org
[11] http://www.globalwordnet.org/AWN
[12] http://dcl.bas.bg/BulNet/general_en.html
[13] http://www.keenage.com
[14] http://www.illc.uva.nl/EuroWordNet
[15] http://borel.slu.edu/lsg
[16] http://olp.dfki.de/ontoselect
[17] http://aims.fao.org/website/AGROVOC-Thesaurus/sub
[18] http://www.ciard.net/partners/labof-chinese-agricultural-ontology-services
[19] http://protege.stanford.edu
[20] A synset is a synonym set: "*a set of words that are interchangeable in some context without changing the truth value of the proposition in which they are embedded*" as defined in WordNet Reference Manual, Princeton University, see http://wordnet.princeton.edu/wordnet/documentation

costly process for domain-specific ontologies. In addition, Ngai et al.'s approach relies on synsets, which are not necessarily evident in formal ontologies.

CLOM via linguistic enrichment: Pazienza & Stellato [41] 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 [43] was developed for the Protégé editor. The plug-in presents an interface to the ontology engineer during ontology development, where word senses (e.g. extracted from WordNet) can be associated with ontological resources. Precision, recall and f-measure are used to measure Pazienta & Stellato's system. Linguistic enrichment of ontological resources may offer useful evidence to the matching techniques. However, as pointed out by the authors themselves, this enrichment process is currently unstandardised. As a result, linguistically enriched ontologies are not vastly available to matching techniques.

CLOM via indirect alignment can be classified as a form of mapping reuse. This is a concept that already exists in MOM [14, p.65]. 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 presented by Jung et al. [25], where indirect alignment is conducted among ontologies in English, Korean and Swedish. Given alignment *A* that is generated between ontology O_1 (e.g. in Korean) and O_2 (e.g. in English), and alignment *A′* that is generated between ontology O_2 and O_3 (e.g. in Swedish), 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 [24] whereby precision and recall are used to measure mapping quality. Assuming the availability of *A* and *A′*, this is a straightforward approach to achieve technically. However, it can be difficult to apply this approach when *A* and *A′* simply do not exist, as CLOM currently remains a challenge in itself.

Translation-based CLOM refers to approaches that are enabled by translations that can be achieved through the use of MT tools, bilingual/multilingual thesauri, dictionaries etc. Typically in translation-based CLOM approaches, a CLOM problem is converted to a MOM problem first, which can then be solved using MOM techniques. Compared to the previously discussed approaches, translation-based CLOM is currently a very popular approach that is exercised by several researchers (discussed next), mostly due to its simplicity of execution and the large number of readily available tools for MT and MOM. Five examples of the translation-based approach to CLOM are discussed next.

The OAEI introduced its first ontology mapping test case involving different natural languages in 2008. The OAEI *mldirectory* test case[21] consists of matching web directories (including Dmoz, Lycos and Yahoo) in different languages (i.e. English and Japanese). Zhang et al. [57] 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[22] tool. It should be noted that among 13 participants in 2008, only one contestant (i.e. RiMOM) submitted results to this test case. These results however were not evaluated by the OAEI[23]. The experience from this test case showed a lack of attention on CLOM at the time, and highlighted the need for further research on mapping techniques in the multilingual environment.

OAEI 2009 introduced the *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)[24]. Among 16 participants, only 2 contestants submitted results (discussed next). Bouma [5] uses 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 were then generated using the GG2WW tool in the monolingual environment. Nagy et al. [33] used DBpedia itself to associate concepts in English and Dutch, since the articles and titles in DBpedia are often labelled in both natural languages. Mappings were then 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, sample matches (some 71-97 matches were randomly selected from 3663 matches generated by GG2WW, and from 2405 matches generated by DSSim) and then evaluated based on a partial gold standard (including 100 reference mappings) using precision and recall[25]. A greater recall was found in the GG2WW tool (around 0.6) comparing to the DSSim tool (around 0.2). However, the 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 widely (between 0.1 to 0.8). Although the evaluation was only partial, 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 higher average precision (0.97) and recall (0.66). It is not known whether this was shown in the GG2WW tool, as it only took part in the VLCR test case.

The VLCR test case was again included in OAEI 2010, where only one tool (RiMOM) took part from a total of 16 contestants. Wang et al. present a record of the number of matches generated by RiMOM in [55] and described an instance-based matching approach at a very high level (it is not clear whether the same translation technique presented in OAEI

[21] The data set is available at http://oaei.ontologymatching.org/2008/mldirectory

[22] http://keg.cs.tsinghua.edu.cn/project/RiMOM/

[23] 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.

[24] The VLCR test case can be found at http://oaei.ontologymatching.org/2009/vlcr/

[25] The evaluation results can be found at http://oaei.ontologymatching.org/2009/results/vlcr/

2008 was used for this test case). However, these matches were never evaluated by the authors. 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[26]. OAEI 2011 does not include any multilingual data sets[27].

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. [54] and Trojahn et al. [51] are discussed next. Wang et al. [54] 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.

A similar approach is presented by Trojahn et al. [51], which incorporates the work presented in [17, 25]. 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 [25]. Trojahn et al.'s approach is evaluated with ontologies in English, French and Portuguese through using precision, recall and f-measure. A range of precision (0.41-0.86), recall (0.05-0.51) and f-measure (0.10-0.62) were achieved.

A common key characteristic shared by translation-based CLOM approaches discussed above is that CLOM is achieved through two steps. Translations of ontology labels are first carried out to overcome the natural language barrier in the given ontologies. This is then followed by MOM techniques. What is evident from this state of the art review is that existing research in CLOM has successfully demonstrated the feasibility of incorporating MT and MOM techniques. However, little effort has been made to investigate the impact of the translations on the subsequent MOM outcome. Furthermore, it is not yet explored whether support can be provided to assist the translation process in order to influence the subsequent mapping outcome. This paper aims to fill this research gap by proposing a configurable cross-lingual ontology mapping system that is able to adjust the mapping outcome by altering the ontology label translations.

2.4 Translations in Cross-Lingual Ontology Mapping vs. Translations in Ontology Localisation

Ontology localisation is defined as "the adaptation of an ontology to a particular language and culture" [48]. 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" [8]. 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, since translation removes the natural language barrier but not necessarily the culture barrier.

An example tool to facilitate the localisation of ontology labels is the LabelTranslator tool [13], which is designed to localise ontologies in English, Spanish and German. The work presented in this paper is different from the LabelTranslator tool as SOCOM++ is a cross-lingual ontology mapping system that uses translations but not necessarily localisations of ontologies. The LabelTranslator tool aims to localise ontology labels (it currently does not provide further localisation support such as making changes to the ontology structure), whereas the work presented in this paper aims to achieve cross-lingual ontology mapping by using translated labels that are not necessarily suitable for localisation. Given rather different goals of the two, the translation requirements for localisation and CLOM thus differ. In a nutshell, translations for the purpose of localisation need to meet the needs of the target community not only through the use of a target natural language, but also adapted to the culture and geopolitical environment of this community. In contrast, translations for the purpose of CLOM need to ensure that the mapping process is able to generate correct mappings by using these translations, which may not have met the localisation requirements.

In summary, a key observation from the review is that, 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. In particular, it is not yet investigated how translation-based CLOM systems can facilitate the generation of high quality mappings by using different translations for the given ontology labels. This paper presents the SOCOM++ system and demonstrates its ability to adjust translations in an effort to configure mapping outcome.

3. SOCOM++ Design

This section presents the design of SOCOM++. As shown in Fig. 2, given ontologies O_1 (in the source natural language) and O_2 (in the target natural language) to be mapped, O_1' (in the target natural language) is generated first by structuring the translations of the O_1 labels according to the original O_1 structure during the *ontology rendition* process. MOM results between O_1' and O_2 (now both in the same target natural language) are then generated by applying *MOM techniques*.

[26] 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

[27] An overview of OAEI 2011 test cases can be found at http://oaei.ontologymatching.org/2011

Finally, CLOM results are generated based on the known MOM results and the O_1 label translations during the *mapping reconstruction* process. The key contribution of SOCOM++ compared to the common translation-based CLOM systems is that SOCOM++ is designed to support the adjustment of the translations selected for the labels in O_1.

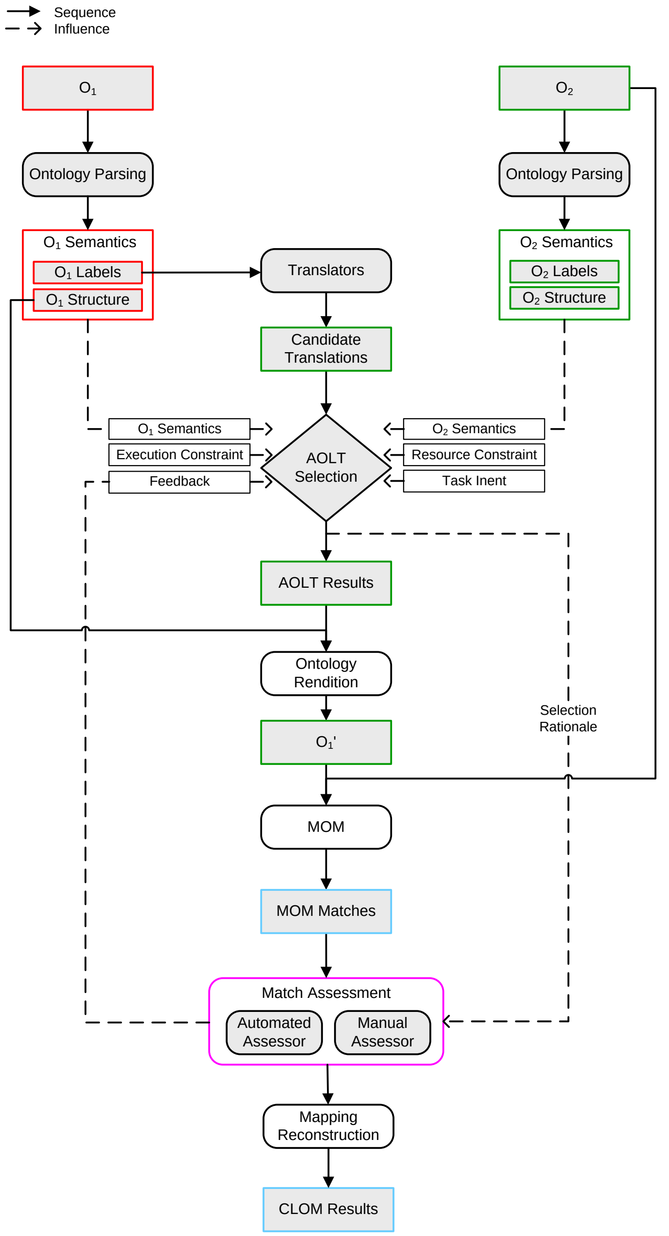


Fig. 2. SOCOM++ Process Diagram. Natural language barrier between O_1 and O_2 are removed given O_1'.

As shown in Fig. 2, there are six configurable inputs to the *AOLT selection* process, including execution constraints, O_1 semantics, O_2 semantics, resource constraints, task intent and feedback. An overview of each input is presented next. For detailed explanation on how each input is used, see section 4.

An execution constraint is a high-level restriction on how the AOLT selection process will proceed. It offers the user the choice between the default configuration and a user-specified configuration of SOCOM++. By having a default configuration, the user can generate initial mappings in a CLOM scenario, analyse the mapping outcome and decide on the further adjustment of SOCOM++.

O_1 semantics refer to the embedded and background semantics of the entities in the source ontology. Similarly, *O_2 semantics* refer to the embedded and background semantics of ontological entities in the target ontology. Embedded semantics refer to formally defined conceptualisations in the ontology such as the semantic surroundings of entities. Background semantics refer to knowledge drawn from external resources such as thesauri. In this paper, 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 are one level higher and/or lower than *C* in the ontological hierarchy. For a property entity *P* (either datatype or object), its surrounding is defined as the entity(ies) which *P* restricts. For an instance *I*, its surrounding node(s) are defined as the class entity(ies) which *I* belongs to. Note 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 order to maintain a distinctive representation for a given entity from another entity in the same ontology, the immediate semantic surroundings are used in SOCOM++. It is a possible direction for future work to determine the optimal or dynamic construction of a node's semantic surroundings.

Resource constraint refers to the availability of external resources (e.g. dictionaries, thesauri). In SOCOM++, this includes the availability of synonyms in the given ontology domain. A lack of synonyms may be evident in some specialised domains whereby there are few other ways to express the same concept, or it may be the case that synonyms are simply not available or 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. generate as many correct matches as possible), or to increase mapping recall (i.e. generate as many matches as possible to ensure the completeness of the mappings).

Feedback aims to improve the matching quality upon recognising how correct matches. By assessing the candidate matches generated in a specific CLOM scenario via an automated assessor (e.g. to infer the correctness of the matches without the involvement of a user) or a manual assessor (e.g. explicit feedback from a user), the system attempts to improve its future selection of the AOLT results based on the selection rationale derived from this assessment. An automated feedback feature is supported in SOCOM++, which is inspired by relevance feedback used in the field of information retrieval (IR). Broadly speaking, there are three types of relevance feedback: explicit, implicit and pseudo feedback [44]. Explicit feedback is obtained after the user issues a query 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. 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 by Duan et al. [11]. SOCOM++ expands on Duan et al.'s work and investigates feedback techniques without the involvement of a user in CLOM scenarios. Assumptions on matches' correctness are based on their confidence levels (generated by MOM tools) in SOCOM++. Although currently there is no obvious method to calculate confidence levels that is a clear success [22], they are however useful indicators as to whether a match is correct or not. In SOCOM++, once feedback is enabled, the system assumes that matches with confidence levels above a user-specified threshold are correct. It then investigates how these matches are generated, i.e. what AOLT selection rationale was used. The rationale is then used to select AOLT results in further iterations of the AOLT selection process.

4. SOCOM++ Implementation

The technical details of SOCOM++ are discussed in this section. In particular, how each input discussed in the previous section are used is explained in this section. To realise the design shown in Fig. 2, a set of system properties is integrated in SOCOM++ that can be configured by the user. Table 1 presents an overview of these properties. Each property is discussed next.

Table 1. User Configurable Properties in SOCOM++. The six configurable inputs (discussed in section 3) are modelled as system properties.

Inputs shown in Design (see Fig. 2)	Implemented Property	Data Type
execution constraint	`defaultAOLTSelection`	`Boolean`
O_1 semantics	`sourceSurrounding`	`Boolean`
O_2 semantics	`targetSurrounding`	`Boolean`
resource constraint	`translationSynonym`	`Boolean`
	`targetSynonym`	`Boolean`
task intent	`correctnessOptimise`	`Boolean`
	`completenessOptimise`	`Boolean`
pseudo feedback	`threshold`	`0.0<Float<1.0`

In SOCOM++, when *execution constraint* (i.e. the `defaultAOLTSelection` property in Table 1) is set to `true`, all other property values are ignored and the default algorithm is executed to select AOLT results. For the AOLT selection process to take other property values into account, execution constraint must be set to `false`.

O_1 semantics (i.e. the `sourceSurrounding` property in Table 1) can be set to `true` to account the semantic surroundings of source entities during the AOLT selection process. Similarly, semantic surroundings of the target entities are taken into account when *O_2 semantics* (i.e. the `targetSurrounding` property) is set to `true`. In SOCOM++, the output from analysing the semantics from O_1 is referred to as *O_1 analysis*, which includes the extracted O_1 labels, their corresponding semantic surroundings (extracted using the Jena Framework 2.5.7[28]), candidate translations (generated using the GoogleTranslate API 0.5[29] and Microsoft Translator API[30]) and synonyms for these candidate translations (generated using the Big Huge Thesaurus API [31] for synonyms in English and synonyms-fr.com for synonyms in French), Similarly, *O_2 analysis* (i.e. output from analysing the semantics from O_2) includes the O_2 labels, their synonyms and corresponding semantic surroundings. Both O_1 semantics and O_2 semantics are written in XML and stored in the eXist database[32].

Resource constraint is modelled by two properties (i.e. `translationSynonym` and `targetSynonym` in Table 1), which offer the option of restricting external resources during the AOLT selection process. When `translationSynonym` is enabled (i.e. value is sent to `true`), synonyms for candidate translations of the source labels are accounted. Similarly, when `targetSynonym` is enabled, the synonyms collected for the target labels are taken into account during the selection process.

Task intent is also modelled by two properties (`correctnessOptimise` and `completenessOptimise` as shown in Table 1), where only one property can be enabled (i.e. set to `true`) at a time. This is because the current implementation attempts to improve either just the correctness or just the completeness of the matches, but not both at the same time. Optimising correctness is achieved by assuming the matches generated from the first iteration with 1.0 confidence levels are correct, analysing how they were achieved, i.e. conclude the selection rationale, and computing a second iteration of the AOLT selection process. Optimising completeness works similarly, except that the assumption assumes all matches (with any confidence level) from the first iteration are correct. Correctness is optimised by eliminating uncertain matches (i.e. any match that has less than 1.0 confidence level) and attempting to increase the number of certain matches (i.e. matches with 1.0 confidence levels), which in turn optimises precision. During this process however, it is possible that correct matches are eliminated (i.e. those matches that have lower than 1.0 confidence levels, but are still correct). Completeness optimisation avoids incorrect eliminations of uncertain matches (since all matches in the first iteration are assumed to be correct), which offers a much more relaxed strategy compared to optimising correctness.

Lastly, *pseudo feedback* (i.e. the `threshold` property in Table 1) is similar to task intent, except that the user is able to specify what assumptions should be made after the first iteration. The value for pseudo feedback can be set to any float value between `0.0` and `1.0`. This feature can thus be considered as an intermediate between the two extremes: optimisation of completeness and optimisation of correctness (as modelled in task intent).

As shown in Fig. 2, *Ontology Rendition* is responsible for converting O_1 to O_1', which is necessary to overcome the natural language barrier between O_1 and O_2. In contrast to existing approaches discussed in section 2.3, the process of generating translated labels in order to construct O_1' is more

[28] http://incubator.apache.org/jena/

[29] http://code.google.com/apis/language/translate/overview.html

[30] http://www.microsofttranslator.com/dev later named as Bing.

[31] http://words.bighugelabs.com/api.php, its data is based on WordNet, the Carnegie Mellon Pronouncing Dictionary and crowd-sourced suggestions.

[32] http://exist.sourceforge.net/

sophisticated in SOCOM++, whereby the translation outcome can be adjusted according to the system configuration (as shown in Table 1). To initiate SOCOM++, a validation is first performed to ensure meaningful values are contained in the property configurations[33].

For a label in O_1, its candidate translations and their synonyms (now in the target natural language and stored in *O_1 analysis*) are each compared to what is stored in *O_2 analysis*. This comparison uses case and space insensitive edit distance, which compares a given character string (i.e. a label) to the character strings (i.e. a set of labels). Note that at this stage in the process, these comparisons are made between strings in the same natural language[34]. A translation record is created from this process, an example is shown in Fig. 3.

```
<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"/>
   …
</AOLTRecord>
```

Fig. 3. An Example AOLT Record. The source label 院所 has four candidate AOLT results: `Organization` is derived from the `BHT` (the Big Huge Thesaurus API) and is of candidate type `2`; `establishment` is also derived from the `BHT` and is of type `4`, and so on.

There can be six types of candidate AOLT results in SOCOM++, ordered in terms of the strongest to the weakest as follows:

- Type 1 denotes a match[35] between a candidate translation (from O_1 analysis) and a target label (from O_2 analysis).
- Type 2 illustrates a match between a synonym of a candidate translation and a target label.
- Type 3 refers to matches found between a candidate translation and a target label's synonym.
- Type 4 represents instances when matches are found between a synonym of a candidate translation and a synonym of a target label.
- When the MT tools agree on the translation for a source label, this is stored as a type 5 AOLT result.
- Type 6 refers to machine-generated candidate translations that differ from one another.

Note that type 6 AOLT candidates can only exist given the absence of a type 5 AOLT candidate, since when a type 5 AOLT candidate is recorded, it implies there are two type 6 AOLT candidates which are in agreement with each other. However, there is no need to record both type 5 and 6 AOLT candidates, as the latter only adds redundant entries to the AOLT record. After the candidate AOLT results are generated, the selection for the final AOLT results begins (this is referred to as the AOLT selection process in the rest of this paper).

It is possible that two (or more) source labels may choose the same string as its AOLT result[36] (this is hereafter referred to as translation collisions), hence, the AOLT selection process must also resolve these collisions. To resolve AOLT collisions, the system determines which entity will keep the colliding term and what alternative AOLT will be given to the other entity. Note that collisions are solved as soon as they are detected, hence it always concerns two entities at a time. Recall there are six types of candidate AOLT results. In the case of a collision, the entity with a preferred type will keep the colliding term as its AOLT result, while the other entity must search for an alternative. This is demonstrated in more detail through the trials shown in sections 6 and 7. Fig. 4 presents an example of the final AOLT results. Once the final AOLT results are determined for all the labels in O_1, O_1' is generated using the Jena Framework, which arranges the translations according to the original structure in O_1.

```
<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#Manag
   er"/>
   <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#Assoc
   iate_Professor"/>
   …
</AOLTSelection>
```

Fig. 4. An Example of AOLT Selection. If a second iteration of the AOLT selection process is configured, `type` and `media` will be used to determine the preferred combinations to use in the second iteration. More details are discussed in section 7.

Next, MOM techniques are applied to generate MOM results between O_1' and O_2 in the alignment format[37] using the Alignment API[38]. Since it is now known how entities in O_1' correspond to the original entities in O_1, the CLOM results are finally constructed based on the MOM results and by simply looking up the relevant AOLT selection for each O_1 entities (as shown in Fig. 4).

5. Evaluation Overview

[33] For example, only one of the properties `correctnessOptimise` and `completenessOptimise` can be set to `true` at a time.

[34] String comparisons shown in this paper are achieved by the LingPipe API using utf-8 encoding, see http://alias-i.com/lingpipe/ index.html. Note edit distance between two strings that are in the same natural language are meaningful. For example, the edit distance between *affect* and *effect* is 1, i.e. it takes one edit operation to turn *a* (in *affect*) into *e* (in *effect*). Another example can be 友谊 (meaning *friendship* in Chinese) and 友好 (meaning *friendly* in Chinese), where the distance between them is 1, i.e. it takes one edit operation to turn one string into the other (by changing 谊 to 好). However, the distance between *affect* and 友好 would be meaningless, hence not supported by most string comparison tools (see footnote 1).

[35] A match in this context refers to a pair of strings that has zero edit distance when the white spaces and character cases are ignored (i.e. case and space insensitive).

[36] An example of translation collision may be: for entities 会议 and 会晤 in O_1, the best available candidate AOLT result derived for 会议 is *meeting*, and the best available candidate AOLT result for 会晤 is also *meeting*. This causes a translation collision since these entities are distinctive of each other and should not have the same translation in O_1'.

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

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

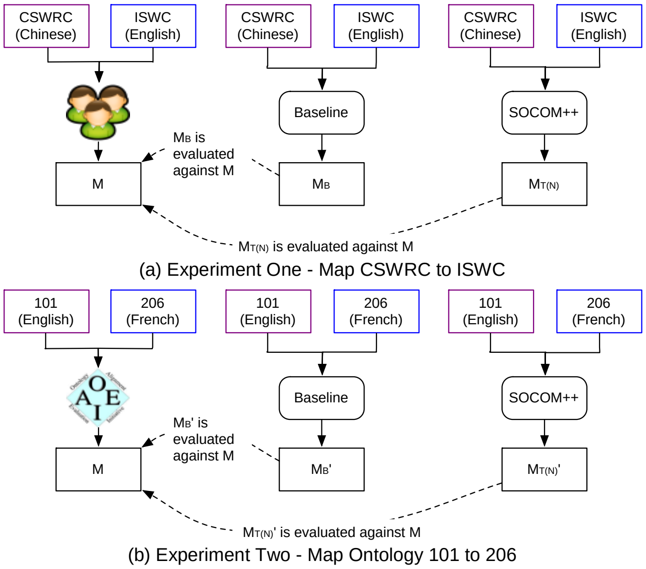
A total of six different configurations of SOCOM++ are applied in this evaluation, as shown in Table 2. This evaluation setup aims to demonstrate that the CLOM outcome can be adjusted by tuning the translations from the AOLT selection process. These six trial configurations are compared against a baseline system, which is representative of the current translation-based CLOM approaches. This baseline system uses the GoogleTranslate API to generate ontology label translations and applies the Alignment API next to achieve mappings. The only distinction between SOCOM++ and the baseline is how the translations of the O_1 labels are achieved. Additional technical details of the baseline system can be found in [17]. Note that the trials presented in this paper are not an exhaustive list of how SOCOM++ can be configured, but rather representative examples of possible adjustments.

Table 2. An Overview of Six SOCOM++ Trial Configurations. Each trial uses a different configuration of the system properties discussed in section 4.

SOCOM++ Configuration AOLT Selection Input	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6
Candidate Translations for O_1 Labels (achieved through the GoogleTranslate API and the Microsoft Translator)	✓	✓	✓	✓	✓	✓
Synonyms of Candidate Translations (achieved through the Big Huge Thesaurus and synonyms-fr.com)	✓	×	✓	✓	✓	✓
O_1 Semantic Surroundings	✓	✓	×	✓	✓	✓
O_2 Labels	✓	✓	✓	✓	✓	✓
Synonyms of O_2 Labels (achieved through the Big Huge Thesaurus and synonyms-fr.com)	✓	×	✓	✓	✓	✓
O_2 Semantic Surroundings	✓	✓	×	✓	✓	✓
2nd Iteration of AOLT Selection	n/a	n/a	n/a	threshold = 1.0	no threshold	threshold = 0.5

Each trial configuration is applied to two CLOM experiments shown in Fig. 5. In experiment one (Fig. 5-a), the Chinese CSWRC ontology[39] is mapped to the English ISWC ontology[40] of the research domain. There are 54 classes, 44 object properties and 30 data type properties in the CSWRC ontology. This CSWRC ontology is manually created based on the English SWRC ontology[41] by a pair of ontology experts (excluding the authors of this paper). The ISWC ontology is of a similar size, containing 33 classes, 17 datatype properties, 18 object properties and 50 individuals. The gold standard[42] in experiment one is generated by a different team of seven mapping experts (excluding the authors of this paper). Experiment two (Fig. 5-b) concerns the mapping of the English 101 ontology[43] to the French 206 ontology[44] of the bibliographic domain. These ontologies and the gold standard[45] are taken from the OAEI 2009 Benchmark test scenario. The 101 ontology contains 36 classes, 24 object properties, 46 data type properties and 137 instances. The 206 ontology is similar to the 101 ontology and contains 36 classes, 23 object properties, 46 data type properties and 137 instances.

In both experiments, eight MOM techniques supported by the Alignment API are applied to generate MOM results for the baseline system and SOCOM++. In experiment one, M (containing 41 exact matches) is the gold standard between the CSWRC ontology and the ISWC ontology. M_B is the matches generated by the baseline system containing eight sets of matches (each set is generated by a MOM algorithm). $M_{T(N)}$ is the matches generated by SOCOM++, where N is the trial number. For example, M_{T1} contains eight sets of matches generated from trial one; M_{T2} contains eight sets of matches generated from trial two, and so on. $M_{T(N)}$ is evaluated against M and compared to M_B. In experiment two, M' (containing 97 exact matches) is the gold standard between ontology 101 and 206. M_B' is the matches generated by the baseline system. $M_{T(N)}$' refers to the matches generated by the SOCOM++, where N is the trial number. $M_{T(N)}$' is evaluated against M' and compared to M_B'.


CSWRC (Chinese)
ISWC (English)
Baseline
SOCOM++
M
M_B
M_T(N)
M_B is evaluated against M
M_T(N) is evaluated against M
101 (English)
206 (French)
O A E I
M_B'
M_T(N)'
M_B' is evaluated against M
M_T(N)' is evaluated against M

(a) Experiment One - Map CSWRC to ISWC

(b) Experiment Two - Map Ontology 101 to 206

Fig. 5. Two CLOM Experiments used to evaluate the Trial Configurations of SOCOM++. These are example scenarios rather than an exhaustive list of possible CLOM scenarios.

6. Three Trials to adjust Ontology Semantics

This section presents three trial configurations (1, 2 and 3) that focus on the adjustment of the inputs related to the ontology semantics. Further iterations of the AOLT selection process (discussed in section 7) are not conducted in these trials. Trial

[39] The CSWRC ontology can be found at http://webhome.csc.uvic.ca/~bofu/SOCOM++/CSWRC.owl

[40] The ISWC ontology can be found at http://annotation.semanticweb.org/ontologies/iswc.owl

[41] http://ontoware.org/swrc/swrc/SWRCOWL/swrc_v0.3.owl

[42] The gold standard can be found at http://webhome.csc.uvic.ca/~bofu/SOCOM++/ref.rdf

[43] The 101 ontology can be found at http://oaei.ontologymatching.org/2009/benchmarks/101/onto.rdf

[44] The 206 ontology can be found at http://oaei.ontologymatching.org/2009/benchmarks/206/onto.rdf

[45] The gold standard can be found at http://oaei.ontologymatching.org/2009/benchmarks/206/refalign.rdf

one (discussed in section 6.1) focuses on adjusting the execution constraint property and illustrates the default AOLT selection in COCOM++. Trial two (discussed in section 6.2) focuses on adjusting the resource constraint property and illustrates a scenario where there is a lack of background resources during the AOLT selection process. Trial three (discussed in section 6.3) focuses on adjusting the embedded semantics such as semantic surroundings.

6.1 Trial One - adjust Execution Constraint

Trial one investigates whether the default AOLT selection process in SOCOM++ can improve the mapping quality compared to the baseline system. The setup is discussed in section 6.1.1 and the findings are discussed in section 6.1.2.

6.1.1 Trial Setup

The default configuration in SOCOM++ makes use of all the resources that are available to assist the AOLT selection process, including the candidate translations for O_1 labels, synonyms of these candidate translations, source semantic surroundings, O_2 labels, their synonyms, target semantic surroundings (see Table 2). For each label in O_1, its candidate translations and synonyms are compared to the data in O_2 analysis and a record of candidate AOLT results are generated as shown previously in Fig. 3.

When selecting the AOLT result for a source label, the system looks through the AOLT record for the lowest possible candidate type. In the absence of a low candidate type for a source label, an alternative candidate AOLT result with a higher type would be selected. In the example shown in Fig. 3, the source label 院所 does not have a type 1 candidate AOLT, then the type 2 candidate would be selected as its AOLT result. Note that more than one candidate AOLT with the same candidate type may exist for a source label, (an example is shown in Fig. 3, the source label 院所 has two type 6 candidate AOLT results), in these situations, the candidate AOLT that is most similar to the target surrounding is chosen as the final AOLT result.

Collision (see footnote 36) resolution strategies used in trial one are summarised in Table 3. The entity with a lower candidate type will keep the colliding term as its AOLT result, and the other entity will seek an alternative from the AOLT record with the next lowest possible candidate type, as shown in collision scenarios i to x. If a pair of collided entities involves the same candidate type, as demonstrated by scenario xi in Table 3, the colliding 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 (via string comparisons) to the candidate AOLT will keep this colliding term as its AOLT result, and the other entity will seek the next available candidate in the same fashion as discussed above. If collisions remain unresolved after all available candidates in the AOLT record have been investigated, a unique integer is selected at random which is attached to the colliding term as the AOLT result for this entity. This technique is designed to allow the system to break out from the recursive process that seeks the next best AOLT result. This breakout approach is used in all trial configurations presented in this paper as well as in the baseline system.

Table 3. Collision Resolution in Trial One. Collisions are always solved between a pair of entities at a given time.

Collision Scenario	Candidate AOLT		Solution
	E_1	E_2	
i	type = 1	type = 2, 3, 4, 5 or 6	E_1 keeps the colliding AOLT; E_2 seeks alternative AOLT with lowest possible candidate type other than the current one.
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 colliding AOLT; E_1 seeks alternative AOLT with the lowest possible candidate type other than the current one.
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 colliding AOLT; the other entity seeks alternative AOLT with the lowest possible candidate type other than the current one.

6.1.2 Findings and Analysis

Precision, recall and f-measure in trial one can be found in the appendix[46]. These diagrams are generated when a match is considered correct as long as it is included in the gold standard regardless of its confidence level.

In experiment one, the improvement on precision is evident across all eight MOM algorithms when applying the trial one configuration compared to the baseline system. The average precision in M_{T1} is 0.4155, which is an improvement of 9.54% compared to the average precision in M_B (at 0.3793). A similar result is found in recall: when the trial one configuration is applied, equal (in the case of the *EditDistNameAlignment* algorithm) or higher (in the case of all other algorithms) recall is found with respect to the baseline system. Particularly in the case of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm, substantially higher recall scores are obtained in this experiment. This is because both algorithms are lexicon-based and use strict string comparison techniques when generating matches[47]. Given AOLT results, recall is thus greatly improved when using these algorithms. An average recall of 0.6488 is found in M_{T1}, which is an improvement of 15.04% compared to M_B (at 0.5640). Higher f-measure is found in all MOM algorithms given the trial one configuration. This suggests that the quality of the matches generated by using the trial one configurations is higher than those generated by the baseline system. On average, an f-measure of 0.4654 is found

[46] In all figures shown in this paper, the eight MOM techniques supported by the Alignment API are represented as follows: 1 represents the *NameAndPropertyAlignment* algorithm, 2 represents the *StrucSubsDistAlignment* algorithm, 3 represents the *ClassStructAlignment* algorithm, 4 represents the *NameEqAlignment* algorithm, 5 represents the *SMOANameAlignment* algorithm, 6 represents the *SubsDistNameAlignment* algorithm, 7 represents the *EditDistNameAlignment* algorithm and 8 represents the *StringDistAlignment* algorithm. These algorithms are representative of state of the art MOM techniques that are often string and structured-based. For further details on how these algorithms generate matches, see [14].

[47] Only matches with 1.0 confidence levels are generated by these algorithm since only entities with identical labels are matched.

in M_{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_{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_{T1} and M_B) and further supports the finding that matches generated using the trial one configuration are of higher quality than those generated by the baseline system in this experiment.

In experiment two, improvement in precision is evident across all eight MOM algorithms. On average, a higher precision of 0.7394 was achieved in trial one compared to 0.6918 that was achieved in the baseline. This is an average improvement of 6.88%. More visible improvement is shown in the recall generated. A mean recall of 0.6057 was found in the baseline, and a higher mean of 0.6261 was found in trial one. This is an average improvement of 3.37%. 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_{T1}'. This is an improvement by 5.31% in the overall quality of the matches generated. The p-value derived from paired t-test is 0.023, which further demonstrates that the difference in the improved mapping quality shown in trial one is statistically significant.

Table 4 shows the evaluation results on confidence levels. In experiment one, increased confidence mean and decreased standard deviation is found in trial one for all MOM algorithms. On average, the mean confidence in M_{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. In experiment two, with the exception of the *SubsDistNameAlignment* algorithm, all other algorithms showed improved confidence levels when using the trial one configuration. The mean confidence of M_B' (at 0.9481) is improved by 0.95% (to 0.9571) in M_{T1}'. The average standard deviation of M_B' (at 0.1207) is decreased by 11.76% in M_{T1}' (to 0.1065). These results show that on average, the matches generated in trial one are more confident with less dispersed confidence levels compared to the baseline. Note that precision, recall, f-measure, confidence mean and standard deviation are used in combination in all evaluations shown in this paper. In addition, statistical significant tests are used to reduce bias and present a holistic view on the matching quality, as each measurement on its own may be misleading.

Table 4. Evaluation Results on Confidence Levels in Trial One. Mean is the average confidence level achieved. Standard deviation measures the dispersion of the confidence levels. High quality matches are those that have high mean and low standard deviation.[48]

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

6.2 Trial Two – adjust Resource Constraint

Trial two investigates the impact that lacks of background semantics (e.g. when thesauri are unavailable to the AOLT process) have upon the matching quality. The setup is presented in section 6.2.1 and the findings are presented in section 6.2.2.

6.2.1 Trial Setup

In specialised domains, it may be the case that there simply is few other ways to express certain concepts, or background resources such as thesauri are simply not available or accessible. Trial two investigates how the mapping outcome is affected given a lack of background semantics.

As discussed previously in section 4, resource constraint is modelled by two properties `translationSynonym` and `targetSynonym`. In trial two, both properties are configured to `false` to illustrate a scenario where thesauri are unavailable[49]. This means that the synonyms of candidate translations for source labels and the synonyms for target labels are not included during the AOLT selection process. As a result, there are only type 1, 5 and 6 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 candidate types.

A summary of the strategies used to resolve collisions in trial two is presented in Table 5. For a pair of collided entities E_1 and E_2, their AOLT results' respective candidate types are checked. The entity with the lower type keeps the colliding term as its final AOLT result, and the other entity seeks an alternative translation, as demonstrated by the collision scenarios i, ii, iii and iv in Table 5. If both entities arrive to the same AOLT result with an equal candidate type (as demonstrated in scenario v), the entity with semantic surrounding that is most similar (i.e. lowest aggregated edit distance) to that of the source label will keep the colliding term as its AOLT result, and the other entity must seek an alternative translation (i.e. with the lowest possible candidate type other than the current one). 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 colliding term for the entity with no more appropriate alternatives.

Table 5. Collision Resolution in Trial Two. The candidate type of the AOLT result and the semantic surroundings are used to resolve collisions.

Collision Scenario	Candidate AOLT		Solution
	E_1	E_2	
i	type = 1	type = 5 or 6	E_1 keeps the colliding AOLT; E_2 seeks alternative AOLT with lowest possible candidate type other than the current one.
ii	type = 5	type = 6	
iii	type = 5 or 6	type = 1	E_2 keeps the colliding AOLT; E_1 seeks alternative AOLT with the lowest possible candiate type other than the current one.
iv	type = 6	type = 5	

[48] Note that not all matching algorithms generate matches with varied confidence levels, such as the *NameEqAlignment* algorithm and the *StringDist-Alignment* algorithm, which only created matches that have a confidence level of 1.0 in the experiments shown in this paper, hence they are not included in the study on confidence levels.

[49] Note that the execution constraint, i.e. `<entry key="default"/>` must be set to `false` for the system to consider the values set in other properties.

v	E_1 type = E_2 type	Entity that is most similar to source surrounding keeps the colliding AOLT; the other entity seeks alternative AOLT with the lowest possible candiate type other than the current one.

6.2.2 Findings and Analysis

Precision, recall and f-measure found in trial two are presented in the appendix. In experiment one, with the exception of the *NameAndPropertyAlignment* algorithm, all other matching algorithms experienced some degree of improvement on precision. On average, M_B achieved a precision of 0.3793, and a higher precision of 0.4437 was achieved by M_{T2}. This is an average improvement of 16.98%. Significant improvement is also evident in recall. An average recall of 0.5640 was found in M_B where as an average of 0.6616 was found in M_{T2}. This is a 17.30% improvement. Overall, an average f-measure of 0.3782 was found in M_B, and an average of 0.4674 was found in M_{T2}. This is an improvement by 23.59%. This finding is further supported by the p-value found in the paired t-test: with a p-value of 0.019, the paired t-test rejects the 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 in M_{T2}'. An average precision of 0.7569 was found in M_{T2}', which is an improvement by 9.41% compared to the baseline system (at 0.6918). The average recall (at 0.6521) is also improved in trial two, which is an improvement by 7.66% compared to the baseline system (at 0.6057). Except for the *NameAndPropertyAlignment* algorithm and the *StringDist-Alignment* algorithm, all other algorithms generated equal or higher recall scores. 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_{T2}'. An average f-measure of 0.6886 was found in trial two, which is an improvement of 8.49% compared to the baseline (at 0.6347). The p-value generated from the paired t-test is 0.006, which supports the statistical significance of the findings so far.

The evaluation results on the confidence levels are shown in Table 6. In experiment one, the confidence means are increased and the standard deviations are decreased for all matching algorithms in M_{T2}. On average, a mean of 0.9326 was found in trial two, which is an improvement by 5.62% compared to the baseline (at a mean of 0.8830). An average standard deviation of 0.1088 was found in M_{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. Results in Table 6 show that matches in M_B' were more confident and with less dispersed confidence levels than matches in M_{T2}'.

Table 6. Evaluation Results on Confidence Levels in Trial Two.

Exp.	MOM Technique	Baseline		SOCOM++ Trial 2 Configuration	
		St. Dev.	Mean	St. Dev.	Mean
1	1	0.1014	0.9374	0.0922	0.9560
	2	0.2505	0.7505	0.2379	0.7752
	3	0.2505	0.7505	0.0404	0.9791
	5	0.0582	0.9649	0.1633	0.9510
	6	0.1618	0.9041	0.1040	0.9431
	7	0.0123	0.9909	0.0150	0.9914
	Avg.	0.1391	0.8830	0.1088	0.9326
2	1	0.0909	0.9674	0.1483	0.9323
	2	0.1509	0.9059	0.2188	0.8295
	3	0.1545	0.9440	0.1237	0.9356
	5	0.1556	0.9431	0.1233	0.9376
	6	0.1541	0.9372	0.2299	0.8664
	7	0.0179	0.9913	0.0173	0.9898
	Avg.	0.1207	0.9481	0.1435	0.9152

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. Consequently, a greater number of matches can be generated, which leads to increased precision, recall and f-measure. Based on this finding, one could then speculate 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.

6.3 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. The setup is discussed in section 6.3.1 and the findings are presented in section 6.3.2.

6.3.1 Trial Setup

As discussed previously in section 4, the embedded semantics of the source ontology is modelled by the property `sourceSurrounding` and the embedded semantics of the target ontology is modelled by the property `targetSurrounding`. In trial three, both properties are set to the value `false`. This configuration thus ignores 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. However, different from trial one, the configuration in trial three does not allow translation collisions to be resolved by comparisons made to semantic surroundings of the ontological resources (since semantic surroundings are ignored in this trial). In trial three, when a collision is detected between two entities E_1 and E_2, their candidate types are checked. The entity with a lower type keeps the colliding term as its AOLT result, and the other entity must seek an alternative. This is discussed previously in trial one (see Table 3, scenario i to x). When entity E_1 and E_2 arrive to the same AOLT result with 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 surrounding. 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 colliding term in the absence of any alternatives.

6.3.2 Findings and Analysis

Precision, recall and f-measure found in trial three are presented in the appendix. 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 precision in 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 in trial three. On average, a precision of 0.3769 was found in M_{T3}, which is a 0.63% decline compared to M_B (at 0.3793). This deterioration is even more evident in recall, where no improvement is shown in any matching algorithm in trial three. At an average recall of 0.4848, this is a fall by 14.04% in M_{T3} compared to M_B (at 0.5640). Consequently, the f-measure generated in M_{T3} is poorer in this trial than in M_B. On average, an f-measure of 0.3457 was found in M_{T3}, which is an 8.59% decrease compared to M_B (at 0.3782). The p-value from paired t-test on the f-measure scores is 0.05, which is the borderline to reject the null hypothesis and suggests that there is a difference between the baseline and trial three.

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_{T3}' than in M_B'. An average precision of 0.7105 was found in trial three, which is a 2.70% improvement from the baseline (at 0.6918). M_{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 in trial three with the exception of the *StringDistAlignment* algorithm. An average recall of 0.6224 was found in M_{T3}', which is a 2.76% improvement compared to M_B' (at 0.6057). Most algorithms generated higher f-measure scores in M_{T3}' in this trial except the *NameEqAlignment* algorithm, the *EditDistNameAlignment* algorithm and the *StringDistAlignment* algorithm. On average, an f-measure of 0.6529 was found in M_{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_{T3}' are higher than those found in M_B' in this trial, which may suggest improved quality in M_{T3}'. However, this is not supported by the paired t-test: with a p-value of 0.148, the null hypothesis cannot be rejected. This finding suggests that although it seems that there is an improvement in f-measure, the difference is not statistically significant. It is therefore difficult to argue that there is a statically significant improvement on the matching quality in this trial.

The results from evaluating the confidence levels are shown in Table 7. In experiment one, the mean confidence is 0.8735 in M_{T3}, which is a 1.08% decrease compared to M_B (at 0.8830). The average standard deviation in M_{T3} is 0.1540, which is a 10.71% increase compared to M_B (at 0.1391). A similar result is found in experiment two. The mean confidence is decreased by 1.70% in M_{T3}' to 0.9320 compared to M_B' (at 0.9481). The standard deviation is increased by 8.04% to 0.1304 in M_{T3}' compared to M_B' (at 0.1207). These findings suggest that there has not been an improvement in the matches' confidence levels in this trial.

Table 7. Evaluation Results on Confidence Levels in Trial Three.

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

In summary, the configurations used in trial three show a much less superior performance, as predicated in the assumption previously. Particularly when dealing with ontologies containing distinct natural language pairs (i.e. experiment one), the trial three configurations prove to be far from desired. Not only have the precision, recall and f-measure not been improved, but the matches are also less confident with more dispersed confidence levels. 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 (that 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 candidate AOLT pool is available.

7. Three Trials to adjust Second Iterations of the AOLT Selection Process

This section presents the three configurations that focus on executing two iterations of the AOLT selection process. Trial four (discussed in section 7.1), five (discussed in section 7.2) and six (discussed in section 7.3) each presents a different method to select the AOLT results in the second iteration of the AOLT selection process.

7.1 Trial Four – adjust Task Intent: Optimising Correctness

The second iteration of the AOLT selection process is achieved by optimising correctness task intent in trial four. The setup of this trial is discussed in section 7.1.1. The findings are presented in section 7.1.2.

7.1.1 Trial Setup

In trial four, optimising correctness is enabled (i.e. setting the `correctnessOptimise` property to `true`, discussed in section 4), whereby the default AOLT selection process (i.e. the configuration used in trial one) is executed in the first iteration. The system then assumes that only the matches (generated after the first iteration) with 1.0 confidence levels are correct and computes the rationale behind these matches (i.e. which AOLT results were used to generate these matches). This rationale is then used to select the AOLT results in the second iteration. An example rationale for the second iteration of the AOLT selection process in trial four is presented in Fig. 6.

```
<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>
```

Fig. 6. An Example Rationale for the Second Iteration of the AOLT Selection Process when using the Optimising Correctness Task Intent. In this example, the rationale is computed for the *SMOANameAlignment* algorithm. A total of 119 matches (stored in the attribute `matches` of the root element) were generated in the first iteration. Among which, 32 matches (stored in the attribute `estimate` of the root element) had confidence levels of 1.0. 16 matches (stored in the attribute `count` of the first Entry element) were generated using AOLT results that were of type 1 (stored in the attribute `type`) and had been agreed by both MT tools (stored in the attribute `media`), which yields a usage of 50% (stored in the attribute `usage`, calculated as count/estimate). Similarly, usages are calculated for all other AOLT rationales (i.e. combination of `type` and `media`) that appeared in the “correct” matches.

In the second iteration, the rationale is treated as a ranked list of AOLT selection strategies, i.e. the higher the usage, the higher ranked a selection strategy. Note that the rationales are generated on a per-MOM-algorithm basis, thus the ranks of the AOLT selection strategies will differ depending on the MOM algorithm applied. In the example shown in Fig. 6, when using the *SMOANameAlignment* algorithm in the second iteration, the candidate AOLT results (which are stored in the AOLT record, see Fig. 3 for an example) with `type="1"` and `media="both"` are the most preferred translations for the O_1 labels. If such candidates do not exist, the AOLT results with `type="1"` and `media="google"` will be selected. In the absence of the above, in third rank, the AOLT results with `type="1"` and `media="bing"` will be selected and so on. When several selection rationales acquire equal usages (e.g. Fig. 6 shows that the last three elements had the same usage as 0.03125), any one of these selection strategies is considered suitable as long as no translation collision is caused.

To solve translation collisions between a pair of entities E_1 and E_2, the entity with the AOLT result derived from a higher ranked selection strategy will keep the colliding term, and the other entity must seek an alternative AOLT result with a lower selection strategy from the AOLT record (e.g. scenario i and ii in Table 8). When both entities choose the same AOLT result with equally ranked selection strategy, the system checks whether alternative AOLT results exist. If alternative AOLT results are only available for one entity, then this entity must seek an alternative whereas the other entity keeps the colliding term (e.g. scenario iii in Table 8). If alternative AOLT results exist for both entities, then the second entity (i.e. after the colliding term has already been stored as an AOLT result for a previous entity) will seek an alternative AOLT result while the first entity keeps the colliding term (see scenario iv in Table 8). When collisions cannot be solved using solutions presented in Table 8 (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 in section 6.1.1).

Table 8. Collision Resolution in Trial Four. When collisions are detected, the AOLT selection process checks the origins of the colliding term and prioritises the higher ranked selection strategy where possible.

Collision Scenario	Candidate AOLT		Solution
	E_1	E_2	
i	Higher rank in TaskIntent (e.g. Fig. 6)	Lower rank in TaskIntent	E_1 keeps the colliding AOLT; E_2 seeks alternative AOLT with lower ranked selection strategy.
ii	Lower rank in TaskIntent	Higher rank in TaskIntent	E_2 keeps the colliding AOLT; E_1 seeks alternative AOLT with lower ranked selection strategy.
iii	Equal rank in TaskIntent, one entity has alternative candidate AOLT results, the other entity has no alternative candidate AOLT.		The entity with no alternative AOLT keeps the colliding AOLT; the other entity seeks alternative AOLT with lower ranked selection strategy.
iv	Equal rank in TaskIntent, both entities have alternative candidate AOLT results.		The first entity keeps the colliding AOLT; the second entity seeks alternative AOLT with lower ranked selection strategy.

7.1.2 Findings and Analysis

Precision, recall and f-measure generated in trial four are shown in the appendix. In experiment one, with the exception of the *SMOANameAlignment* algorithm and the *SubsDistNameAlignment* algorithm, all other algorithms achieved higher precision in M_{T4}. The improvement is particularly evident in the case of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm, where a precision score of 1.0 had been achieved. On average, a precision of 0.4497 was generated in M_{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. Similar results can be seen in the recall scores generated. On average, a recall of 0.6677 was found in M_{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_{T4}, which is an improvement by 26.92% compared to M_B (at 0.3782). However, the p-value generated from paired t-test is 0.06, which suggests that there is not enough evidence to conclude a difference in f-measure between the two systems in this trial. 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_{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 trial two). Except the *NameAndPropertyAlignment* algorithm, recall is improved for all other algorithms in M_{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_{T4}', which is an improvement by 8.59% on M_B' (at 0.6347).

The p-value generated from paired t-test is 0.01, suggesting the statistical significance of the findings in this experiment.

Table 9 presents the evaluation results on confidence levels. In experiment one, matches in M_{T4} are more confident with less dispersed confidence levels. A mean confidence of 0.9472 was found in M_{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_{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_{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_{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_{T4}', which is an improvement by 2.07% compared to M_B' (at 0.1207).

Table 9. Evaluation Results on Confidence Levels in Trial Four.

Exp.	MOM Technique	Baseline		SOCOM++ Trial 4 Configuration	
		St.Dev.	Mean	St.Dev.	Mean
1	1	0.1014	0.9374	0.0615	0.9830
	2	0.2505	0.7505	0.2472	0.7479
	3	0.2505	0.7505	0.0390	0.9900
	5	0.0582	0.9649	0.0390	0.9900
	6	0.1618	0.9041	0.1083	0.9730
	7	0.0123	0.9909	0.0040	0.9992
	Avg.	0.1391	0.8830	0.0832	0.9472
2	1	0.0909	0.9674	0.1166	0.9598
	2	0.1509	0.9059	0.1816	0.8904
	3	0.1545	0.9440	0.1050	0.9532
	5	0.1556	0.9431	0.1048	0.9548
	6	0.1541	0.9372	0.1835	0.9132
	7	0.0179	0.9913	0.0178	0.9903
	Avg.	0.1207	0.9481	0.1182	0.9436

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 (see higher precision, recall and f-measure from both experiments in trial four compared to trial one). However, there is a trade-off on the confidence levels - in both experiments, lower confidence level means and higher standard deviations were found. Nevertheless, the trial four configuration did improve precision, which was the goal of this setup. Motivated by this result, the optimising completeness task intent is investigated and discussed next.

7.2 Trial Five – Optimising Completeness

Trial five investigates the optimising completeness task intent in SOCOM++. The setup of this trial is presented in section 7.2.1. The findings are discussed in section 7.2.2.

7.2.1 Trial Setup

When the optimising completeness task intent is enabled (i.e. setting the `completenessOptimise` property to `true`, discussed in section 4), two iterations of the AOLT selection process are executed. Different from trial four, the system assumes that all matches (with any confidence levels) are correct for a MOM algorithm in trial five. In other words, trial five does not discard any match generated from the first iteration when generating the selection rationale for the second iteration, since a match may still be correct even though it has lower than 1.0 confidence level. An example rationale generated from using the optimising completeness task intent is shown in Fig. 7. In trial five, translation collisions are solved in the same fashion as trial four (see Table 8).

```
<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"/>
   …
</TaskIntent>
```

Fig. 7. An Example Rationale for the Second Iteration of the AOLT Selection Process when using the Optimising Completeness Task Intent. In this example, the analysis is computed for the *SMOANameAlignment* algorithm. In the first iteration, 119 matches were generated, and all of them are assumed to be correct (see attribute values in the root element). 24 matches used AOLT results of `type="6"` and `media="google"`, 20 matches used AOLT results of `type="5"` and `media="both"`, and so on.

7.2.2 Findings and Analysis

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

In experiment two, with the exception of the *NameEqAlignment* algorithm and the *StringDistAlignment* algorithm, all other algorithms generated higher precision in M_{T5}'. An average precision of 0.7288 was found in M_{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_{T5}'. An average recall of 0.6379 was found in M_{T5}', which is a 5.32% improvement from M_B' (at 0.6057). This is not the highest mean recall that has been achieved in this experiment to date, as the average recall achieved in trial two and trial four are both higher. This finding suggests that trial five 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_{T5}', which is an improvement by 5.80% compared to M_B' (at 0.6347). The p-value (at 0.004) further validates the statistical significance of the results above.

Table 10 presents the evaluation results on the confidence levels. In experiment one, a mean confidence of 0.9252 and an average standard deviation of 0.0973 were found in M_{T5}. This is an average increase by 4.78% on the mean confidence and a decrease by 30.05% on the standard deviation compared to M_B. This finding suggests that the matches generated using the trial five configuration were more confident with less dispersed confidence levels in experiment one. In experiment two, a mean confidence of 0.9441 and an average standard deviation of 0.1205 was found in M_{T5}'. This is an average 0.17% improvement on standard deviation, but a 0.42% decrease on mean confidence. This finding suggests that the matches generated using the trial five configuration may have less dispersed confidence levels, but their confidence means are not quite as high in experiment two.

Table 10. Evaluation Results on Confidence Levels in Trial Five

Exp.	MOM Technique	Baseline		SOCOM++ Trial 5 Configuration	
		St.Dev.	Mean	St.Dev.	Mean
1	1	0.1014	0.9374	0.0943	0.9597
	2	0.2505	0.7505	0.2336	0.7355
	3	0.2505	0.7505	0.0507	0.9734
	5	0.0582	0.9649	0.0507	0.9734
	6	0.1618	0.9041	0.1405	0.9189
	7	0.0123	0.9909	0.0141	0.9904
	Avg.	0.1391	0.8830	0.0973	0.9252
2	1	0.0909	0.9674	0.1079	0.9619
	2	0.1509	0.9059	0.1600	0.9022
	3	0.1545	0.9440	0.1061	0.9525
	5	0.1556	0.9431	0.1498	0.9422
	6	0.1541	0.9372	0.1815	0.9151
	7	0.0179	0.9913	0.0177	0.9905
	Avg.	0.1207	0.9481	0.1205	0.9441

In summary, trial five has successfully demonstrated the optimising completeness task intent when working with ontologies containing distinct natural language pairs (i.e. experiment one). However, this configuration was not as successful when dealing with ontologies with similar natural language pairs (i.e. experiment two). The effectiveness of the trial five configuration is evident through the increased recall generated in both experiments compared to the default configuration (i.e. trial one). However, there is a trade-off on confidence levels, as decreased confidence means and increased standard deviations were found in both experiments.

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 the AOLT selection process, whereby 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. all matches generated in the first iteration are assumed to be correct). Obviously, there can be many other cut-off points in-between. This is investigated in trial six through the use of pseudo feedback.

7.3 Trial Six – adjust Pseudo Feedback

Trial six focuses on pseudo feedback, which offers the user flexible cut-off points (as opposed to the fixed cut-off points in trial four and trial five) when assessing the correctness of initial matches for further iterations of the AOLT selection process. The setup of trial six is discussed in section 7.3.1. The findings are presented in section 7.3.2.

7.3.1 Trial Setup

As discussed in section 4, pseudo feedback 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 `threshold` property to a value that is between 0.0 and 1.0 (see Table 1). In trial six, the threshold for pseudo feedback is set to 0.5. This is the most interesting point between 0.0 and 1.0, since equal to or greater than 0.5 indicates an incline towards confident matches, and less than 0.5 indicates an incline towards not confident matches. Note that trial six does not attempt to present an exhaustive list of all possible cut-off points (since it can be any value between 0.0 and 1.0), or to establish the best possible cut-off point for the two experiments (as that will require extensive trials on various thresholds which will lead to an exhaustive list). Trial six is an example of applying pseudo feedback in SOCOM++.

In trial six, two iterations of the AOLT selection process are conducted, whereby it is assumed that any match generated from the first iteration with confidence level that is equal to or greater than 0.5 is correct. Based on this assumption, a set of selection rationale is computed for the second iteration of the AOLT selection process. Similar to trial four and five, selection rationale are generated on a per-MOM-algorithm basis, and translation collisions are solved in the same way (see Table 8). An example rationale generated in trial six is shown in Fig. 8.

```
<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>
```

Fig. 8. An Example Rationale for the Second Iteration of the AOLT Selection Process when using pseudo feedback. The example is generated for the *SMOANameAlignment* algorithm when the value for pseudo feedback is set to 0.5 (see the attribute values in the root element).

7.3.1 Findings and Analysis

Precision, recall and f-measure generated in trial six are presented in the appendix. In experiment one, with the exception of the *NameAndPropertyAlignment* algorithm, all other algorithms generated higher precision in M_{T6}. An average precision of 0.4462 was found in M_{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_{T6} which is a 33.00% increase compared to M_B (at 0.5640). A similar finding is shown in f-measure. An average f-measure of 0.5062 was found M_{T6}, which is an increase by 33.84% compared to M_B (at 0.3782). The paired t-test also supports the statistical significance of the findings (with a p-value of 0.011).

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

The evaluation results on confidence levels are presented in Table 11. In experiment one, more confident and less dispersed matches were found in M_{T6}. An increased mean confidence by 32.42% (at 0.9310) and a decreased average standard deviation by 5.44% (at 0.0940) were found in M_{T6} compared to M_B. In experiment two, the matches in M_{T6}' contained less dispersed confidence levels, however, are less confident on average compared to M_B'. A decrease 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_{T6}' compared to M_B'.

Table 11. Evaluation Results on Confidence Levels in Trial Six.

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

In summary, trial six has improved the precision, recall and f-measure in both experiments compared to trial one (i.e. default configuration without a second iteration of the AOLT selection process). However, the trade-offs on confidence levels are evident (i.e. increased standard deviation and decreased mean confidence in trial six compared to trial one). This trade-off was shown previously in both trial four and trial five, which suggests that adding iterations of the AOLT selection process is effective at improving precision, recall and f-measure of the matches, but is less effective at improving the confidence levels.

8. Summary of Findings from Six Trials

It is recognised that the experiments shown in this paper are somewhat limited in their domains and natural languages 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 approach used by SOCOM++. Table 12 shows the ranks achieved by all configurations in both experiments. Assuming precision, recall, f-measure, confidence level mean and standard deviation are as equally important as one another, the average rank that is achieved by each trial configuration can be calculated. The highest ranked configuration was trial six (with an average rank of 2.3), and the worst configuration was trial three (with an average rank of 7.4). Note that although trial one and five both achieved an average rank of 3.6, trial five is considered better than trial one as it contains a better rank record, i.e. trial five has a better record since its ranks are in fifth place or higher, whereas trial one lands in sixth rank twice.

Table 12. An Overview of the Ranks achieved by Each Trial Configuration in Both CLOM Experiments. The table orders the trial configurations with respect to their average ranks. For example, trial four achieved rank two 5 times; rank three 3 times; rank four 1 time and rank six 1 time, thus is shown in the second row with an average rank of 2.9.

Rank / Configuration	1	2	3	4	5	6	7	8	Avg. Rank
Trial 6	×4	×1	×3	×2	-	-	-	-	2.3
Trial 4	-	×5	×3	×1	-	×1	-	-	2.9
Trial 5	×2	×1	-	×3	×4	-	-	-	3.6
Trial 1	×3	×1	-	×1	×3	×2	-	-	3.6
Trial 2	-	×1	×3	×3	×1	-	-	×2	4.4
Baseline	-	-	×1	-	-	×1	×5	×3	6.8
Trial 3	-	-	-	-	-	×1	×4	×5	7.4

Five key conclusions can be drawn from analysing the results shown in Table 12. First, the trials that carried out a second iteration of the AOLT process (i.e. trial four, five and six) achieved better rankings than those that did not (trial one, two and three). This finding suggests that using a form of feedback for the AOLT selection 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 selections of AOLT results which consequently leads to better mapping quality.

Secondly, among the trials that did execute a second iteration of the AOLT selection process, trial six can be considered as the best configuration in the experiments conducted. This finding suggests that the approach taken by trial six is better at concluding selection rationales for the second iteration of the AOLT selection process. Compared to trial four, which applies a highest possible cut-off point and trial five, which applies no cut-off point, trial six is relaxed yet effective by applying the 0.5 cut-off point. This assumption is shown to be more useful in the experiments when selecting 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 values of the matches' correctness.

Thirdly, trial three ignores the semantic surroundings during the AOLT selection process and is ranked last as shown in Table 12. This finding in fact further substantiates evidence for the AOLT concept (i.e. translations for the purpose of CLOM should be semantic-oriented in the mapping context, and not take place in isolation of the ontologies involved). The comparison between trial one and trial three clearly demonstrates the drawback of ignoring semantic surroundings. This finding is even more evident when trial three is compared to trial two and the baseline: although a larger candidate translation pool was available in trial three, the mapping quality was still reduced. In other words, increased candidate translation pool implies increased probabilities of choosing inappropriate translations during the AOLT selection process. To overcome this challenge, semantic surroundings need to be included in the AOLT selection process (as demonstrated in trial one).

Fourthly, with the exception of trial three, all other trials have shown higher matching quality compared to the baseline system. Since the only difference between these trials and the baseline system is how the ontology label translations are achieved, this finding demonstrates that AOLT results are more suitable for translation-based CLOM systems.

Last but not least, the trials shown in this paper have successfully demonstrated that the CLOM outcome is adjustable depending on the translations selected for the ontology labels. In conclusion, the SOCOM++ designed to support this adjustment process is shown to be effective in the experiments, where various CLOM outcomes have been generated given the same pair of ontologies.

9. Conclusions & Future Work

Addressing multilinguality is recognised as one of the pressing challenges for the semantic web [3]. Cross-lingual ontology mapping is a relatively unexplored area compared to monolingual ontology mapping, this paper is among the initial efforts in this research field. The key contribution of this research is the concept of configuring appropriate ontology label translations to adjust the mapping quality from a translation-based cross-lingual ontology mapping process. The adjustable mapping outcome is successfully demonstrated through the evaluation of the six trial configurations of SOCOM++. The research shown in this paper is the first attempt that focuses on improving CLOM quality through tuning the intermediate translation outcome. This research has also opened up several research opportunities for future work, discussed next.

Evaluation: the experiments shown in this paper include three natural languages, which is a relatively small sample size. Additional CLOM experiments with more ontology pairs involving additional domains and natural languages will give further insight into the use of the translation selection process in CLOM. Also, the proposed SOCOM++ system can be evaluated through other approaches such as task-oriented approaches such as [39] or end-to-end strategies such as [20].

Implementation: the improvements are shown in a variety of MOM 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 given other MT and MOM tools. Thus, further experiments are necessary. In addition, future 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. Moreover, the MT tools and thesauri shown in this paper cover a general domain of interest, if given for instance biomedical ontologies, more specialised tools may be required. This limitation may be addressed by extending the current implementation.

Other approaches to CLOM: the current translation-based approach to CLOM shown in SOCOM++ is heavily conditioned upon the translation outcomes to generate desired mappings. This approach tailors the selection of the translations to suit specified MOM techniques. In any CLOM scenario however, there will always be a finite set of candidate translations to select from. Though this could be a very large pool, nevertheless, it remains limited. In other words, as long as the CLOM process requires identifying the precise translations for each ontology labels in O_1 (i.e. require the very existence of O_1'), the mapping outcome will be restricted to a finite set of possible translation outcomes which in turn restricts the improvement that can be seen in any given CLOM scenario. Other approaches to CLOM that do not rely on generating 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 community support. 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 used in 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 the community would help to foster innovations in this field.

Acknowledgement

This research is partially supported by the Science Foundation Ireland as part of the Centre for Next Generation Localisation (Grant 07/CE/I1142, www.cngl.ie) and the Federated, Autonomic End to End Communications Services Strategic Research Cluster (Grant 08/SRC/I1403, www.fame.ie) at Trinity College Dublin.

References

[1] 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

[2] 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

[3] Benjamins V.R., Contreras J., Corcho O., Gómez-Pérez A., Six Challenges for the Semantic Web. Semantic Web Workshop, held at 8th International Conference on Principles of Knowledge Representation and Reasoning, 2002

[4] Berners-Lee T., Hendler J., Lassila O., The Semantic Web. Scientific American, May 2001, pp. 29-37, 2001

[5] 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

[6] 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

[7] Chang C., Lu W., The Translation of Agricultural Multilingual Thesaurus, in Proceedings of the 3rd Asian Conference for Information Technology in Agriculture, 2002

[8] 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

[9] 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

[10] 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

[11] 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

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

[13] 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

[14] Euzenat J., Shvaiko P., Ontology Matching. Springer Heidelberg 2007

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

[16] 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

[17] 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

[18] 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, 2011

[19] Gruber T., A Translation Approach to Portable Ontologies. Knowledge Acquisition 5(2):199-220, 1993

[20] 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

[21] 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

[22] 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

[23] Ide N., Véronis J., Word Sense Disambiguation: The State of the Art. Computational Linguistics, 1998 24 (1)

[24] 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

[25] 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

[26] Jurisica, I., Mylopoulos, J., Yu, E.S.K.: Ontologies for Knowledge Management: An Information Systems Perspective. Knowl. Inf. Syst.(2004) 380-401

[27] Kalfoglou Y., Schorlemmer M., Ontology Mapping: the State of the Art. The Knowledge Engineering Review, Vol. 18:1, pp.1-31, 2003

[28] Koehn P., Europarl: A Parallel Corpus for Statistical Machine Translation, MT Summit, 2005

[29] 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

[30] 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

[31] 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

[32] Maedche, A. and Staab, S.: Ontology Learning for the Semantic Web. IEEE Intelligent Systems. 16(2), 72-79, 2001. Special Issue on Semantic Web

[33] 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

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

[35] 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

[36] 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

[37] 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

[38] 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

[39] Noy N.F., Musen M.A., Evaluating Ontology-Mapping Tools: Requirements and Experience. OntoWeb-SIG3 Workshop, Siguenza, Spain, 2002

[40] O'Sullivan D., Wade V., Lewis D., Understanding as We Roam. IEEE Internet Computing, 11(2), pp. 26-33, 2007

[41] 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

[42] 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

[43] 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

[44] 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

[45] 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

[46] Sosnovsky S., Gavrilova T., Development of Educational Ontology for C-programming, International Journal "Information Theories and Applications" Volume 13, pp. 303 – 308, 2006

[47] Srinivasan P., Thesaurus Construction. Data Structures and Algorithms, Prentice-Hall, 1992

[48] 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.

[49] 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

[50] 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

[51] 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

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

[53] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner S., Ontology-Based Integration of Information – A Survey of Existing Approaches. In Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), pp. 108–117, 2001

[54] 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

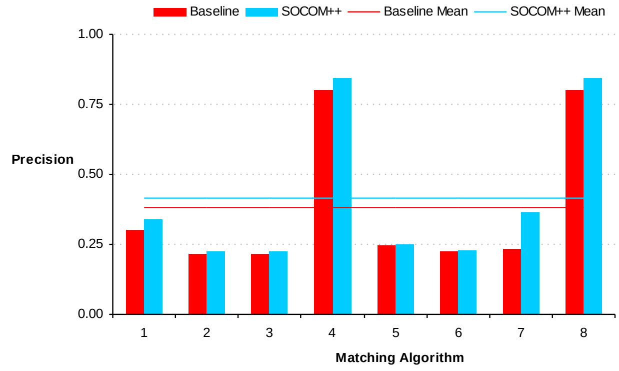
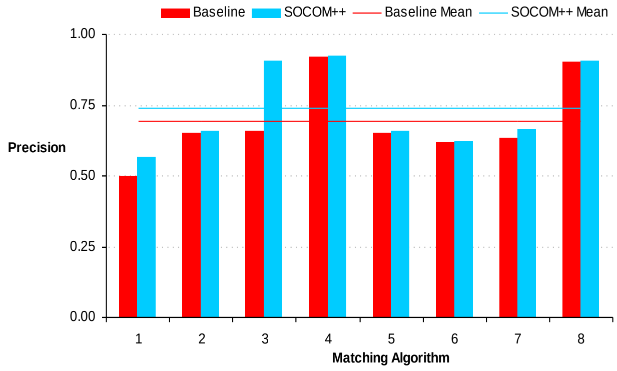
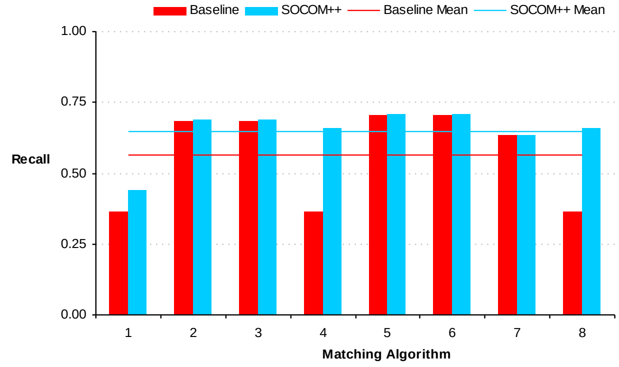
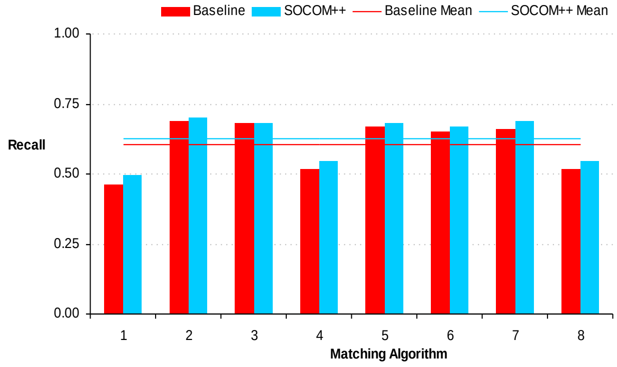
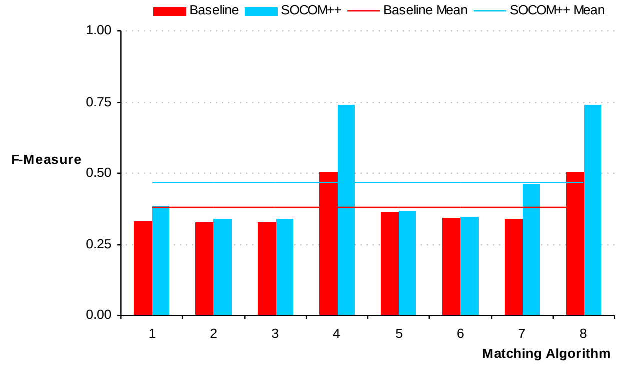
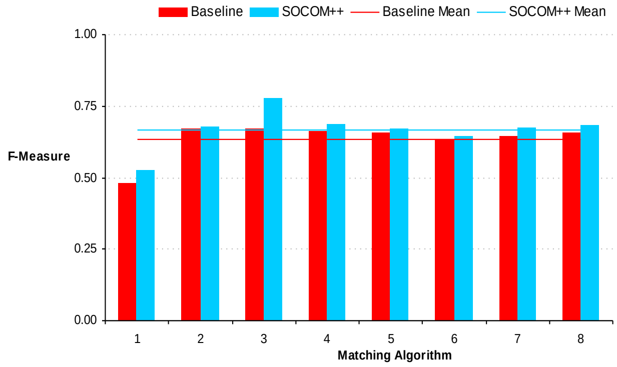
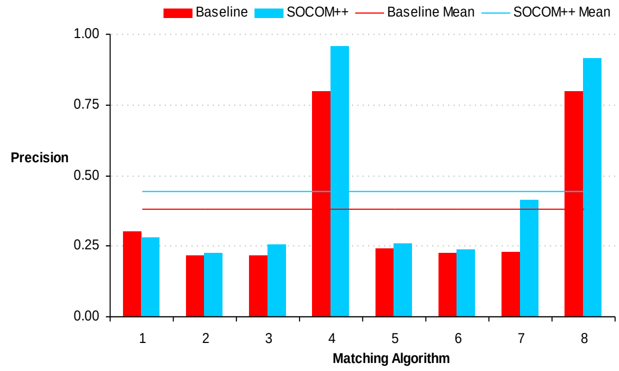
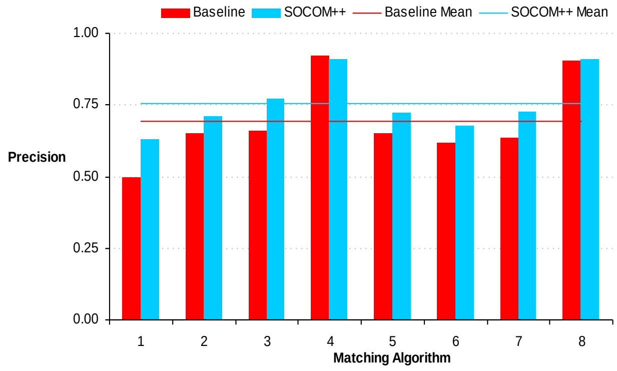
[55] Wang Z., Zhang X., Hou L., Zhao Y., Li J., Qi Y., Tang J., RiMOM Results for OAEI 2010. In Proceedings of the 5th International Workshop on Ontology Matching, CEUR-WS Vol-689, pp. 195-202, 2010

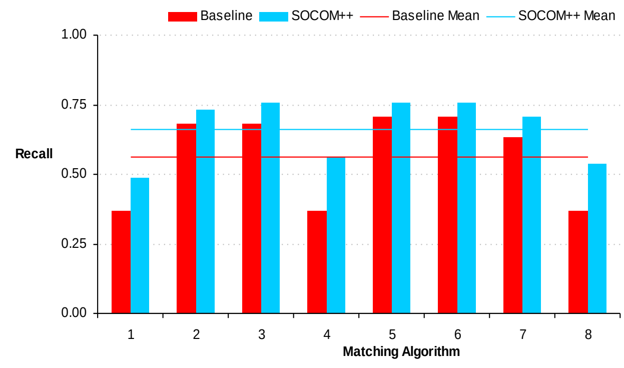
[56] Zhang Z., Zhang C., Ong S. S., Building an Ontology for Financial Investment, in Intelligent Data Engineering and Automated Learning - IDEAL 2000, Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference, pp. 308-313, 2000

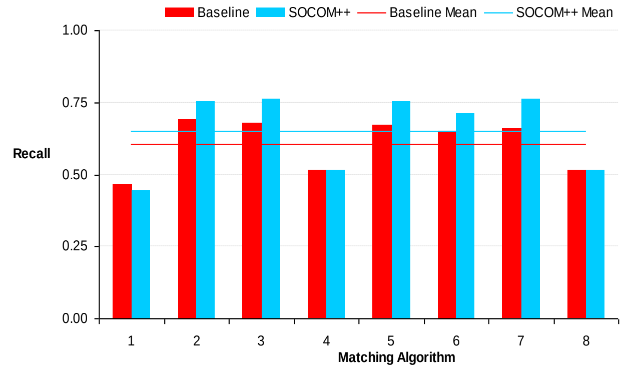
[57] Zhang X., Zhong Q., Li J., Tang J., Xie G., Li H., RiMOM Results for OAEI 2008. In Proceedings of the 3rd International Workshop on Ontology Matching, pp. 182-189, 2008

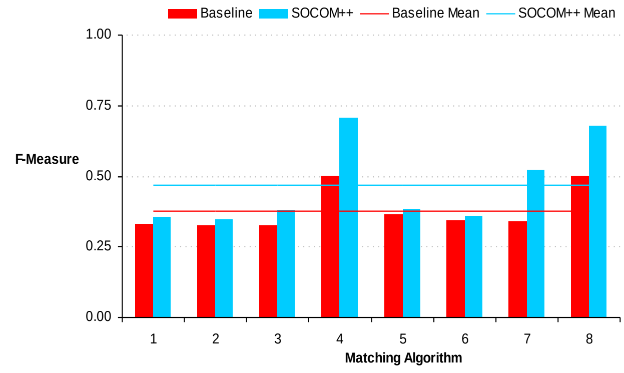
Appendix. Precision, Recall and F-Measure from the Trial Configurations

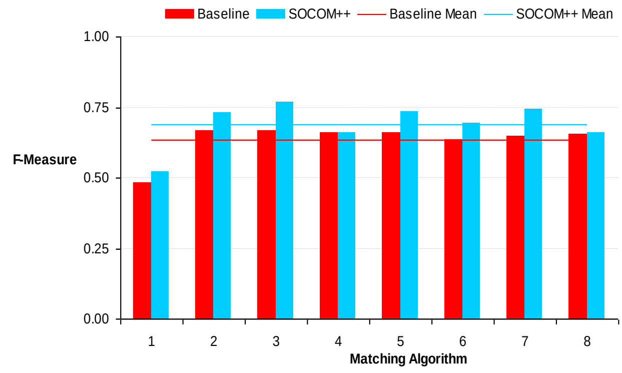
In all figures shown in the appendix, the MOM techniques used in the experiments are presented on the x-axis, the precision, recall and f-measure values are presented on the y-axis.

Trial	Experiment One	Experiment Two
1		
		
		
2		

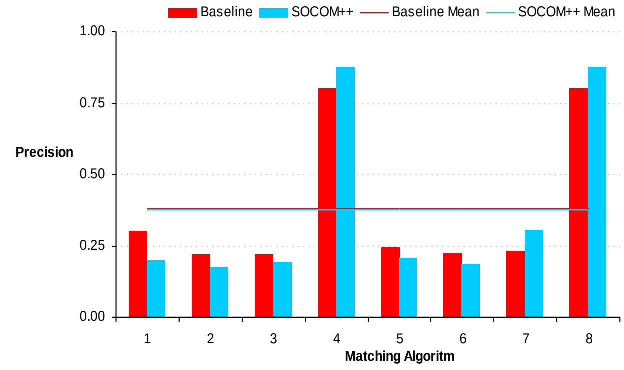

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

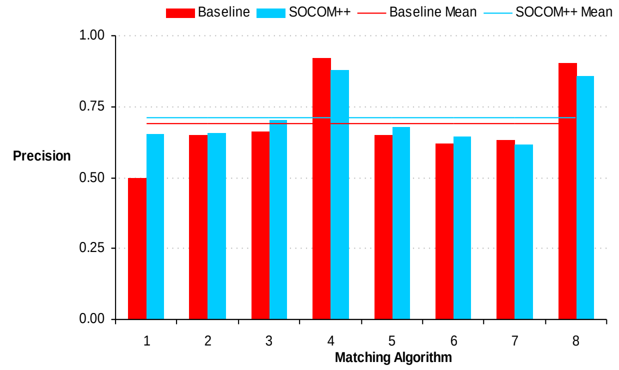

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

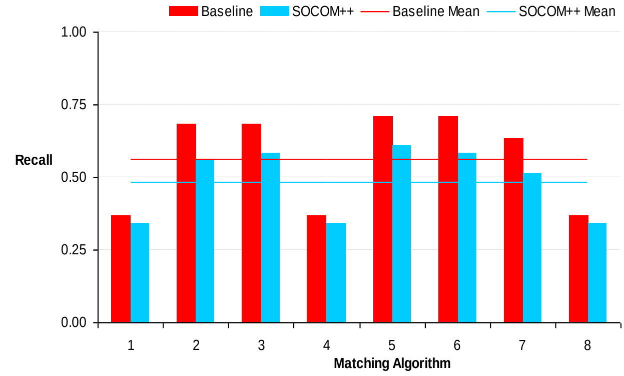

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
F-Measure
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

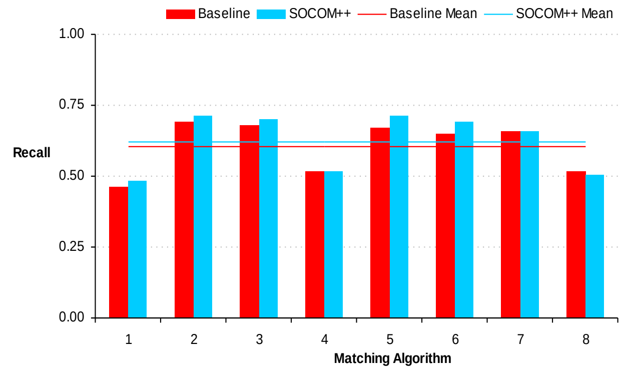

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
F-Measure
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

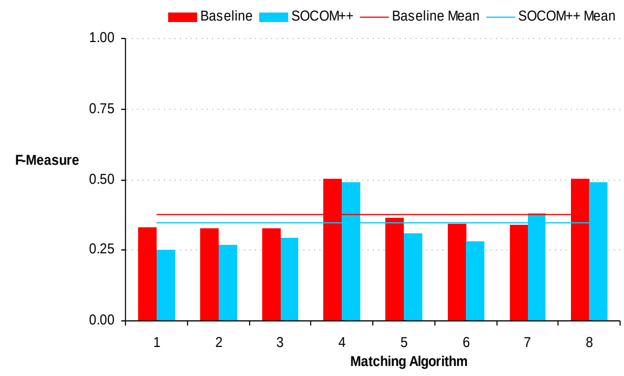
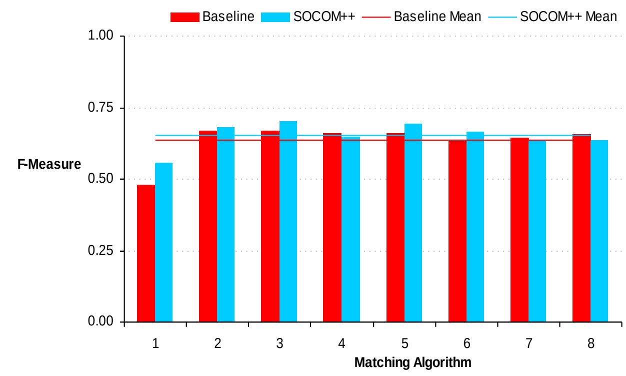
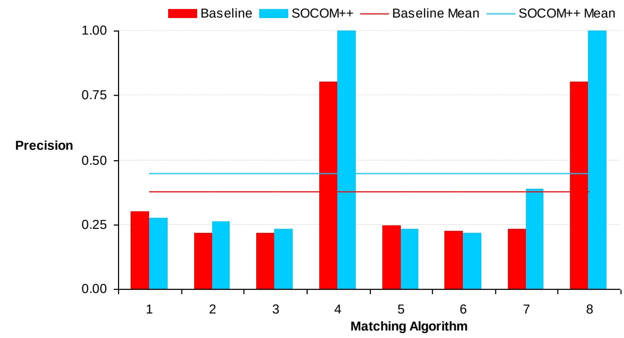
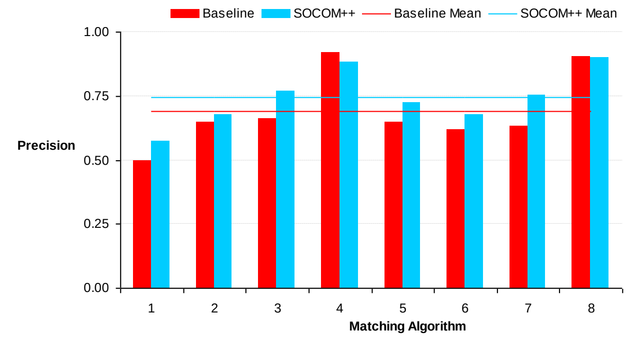
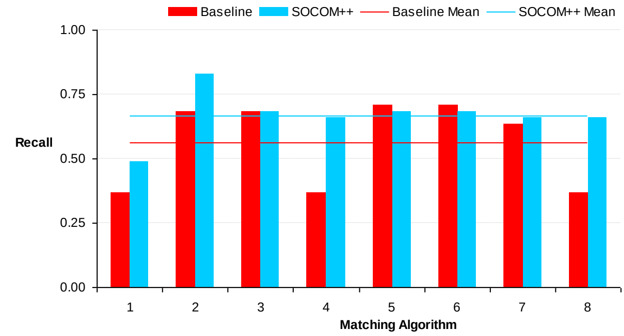
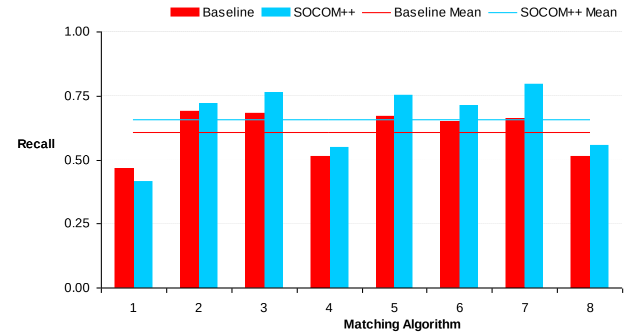
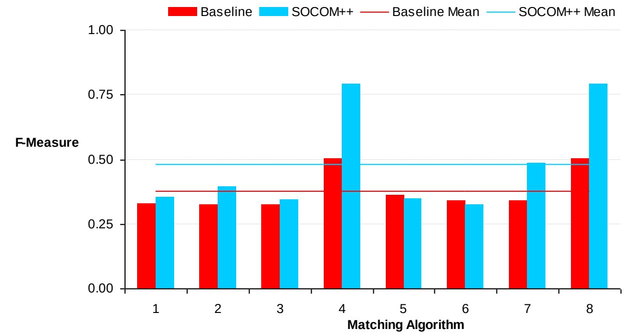
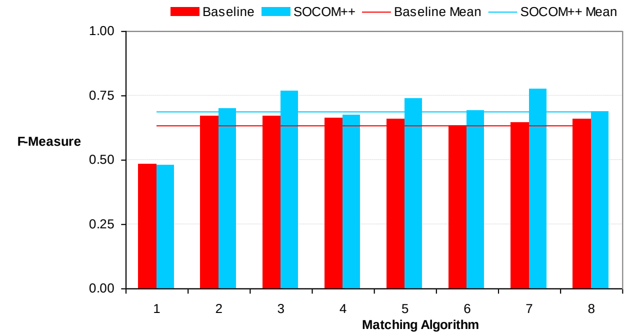
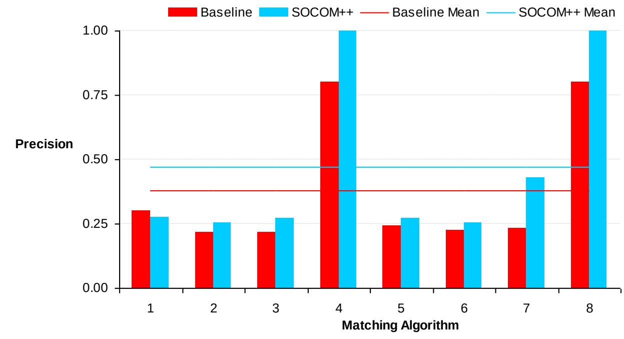
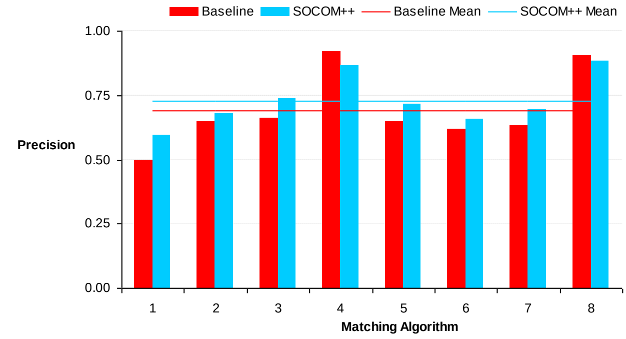
3

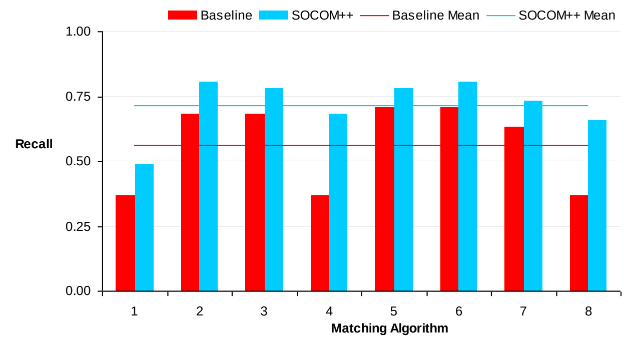

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Precision
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algoritm

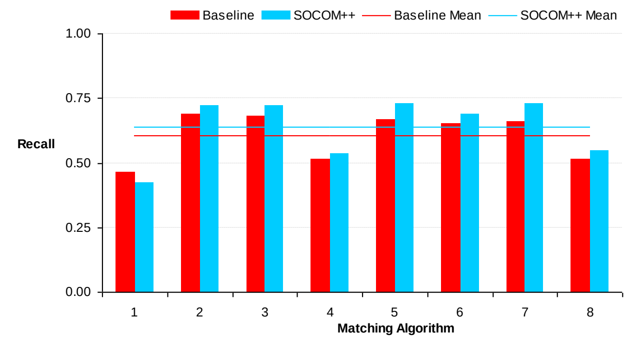

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Precision
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

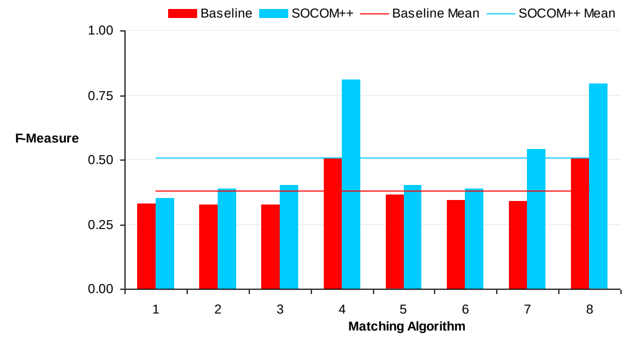

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

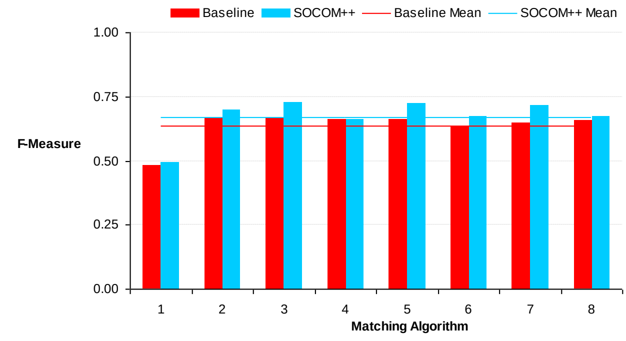

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 F-Measure 1 2 3 4 5 6 7 8 Matching Algorithm	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 F-Measure 1 2 3 4 5 6 7 8 Matching Algorithm
4	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 Precision 1 2 3 4 5 6 7 8 Matching Algorithm	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 Precision 1 2 3 4 5 6 7 8 Matching Algorithm
	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 Recall 1 2 3 4 5 6 7 8 Matching Algorithm	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 Recall 1 2 3 4 5 6 7 8 Matching Algorithm
	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 F-Measure 1 2 3 4 5 6 7 8 Matching Algorithm	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 F-Measure 1 2 3 4 5 6 7 8 Matching Algorithm
5	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 Precision 1 2 3 4 5 6 7 8 Matching Algorithm	Baseline SOCOM++ Baseline Mean SOCOM++ Mean 1.00 0.75 0.50 0.25 0.00 Precision 1 2 3 4 5 6 7 8 Matching Algorithm

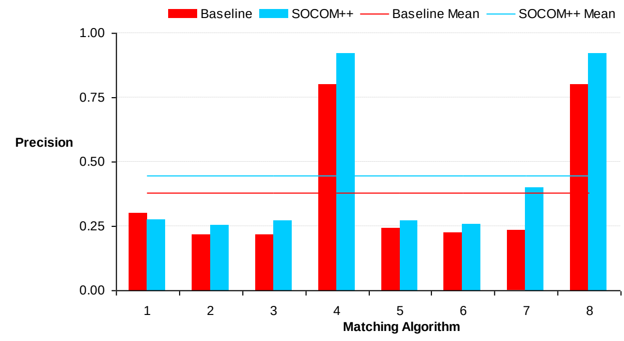

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

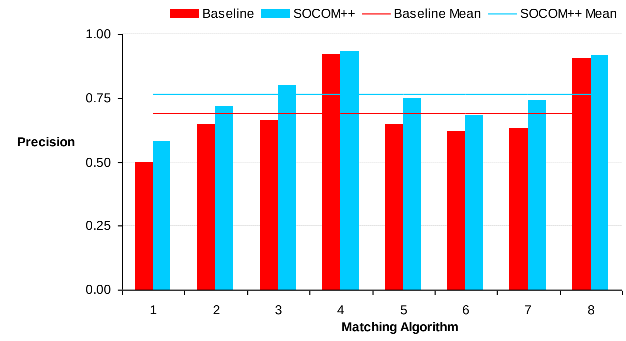

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

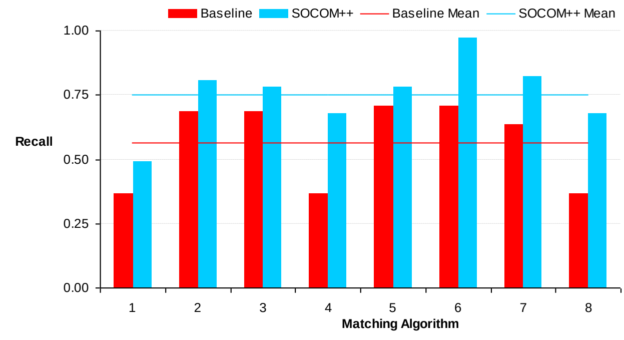

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
F-Measure
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

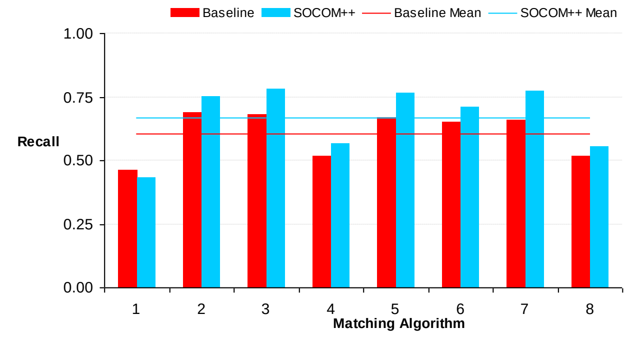

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
F-Measure
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm

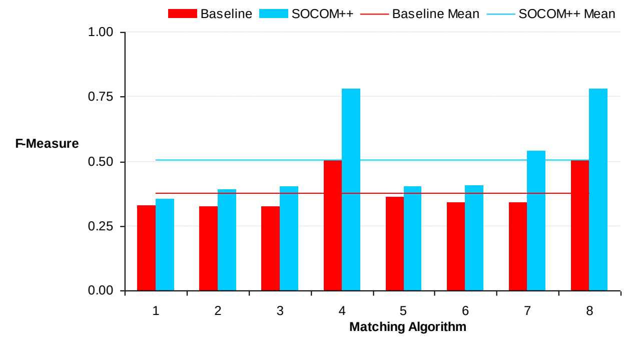
6

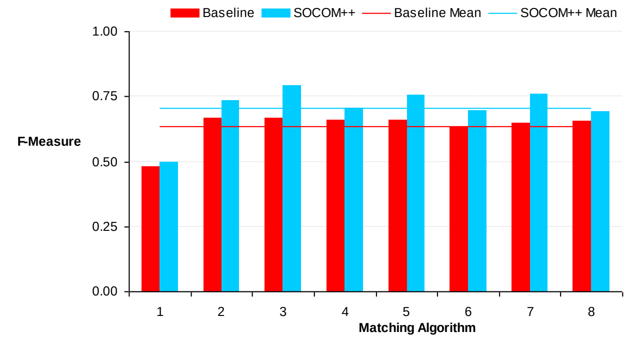

Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Precision
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm


Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Precision
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm


Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm


Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
Recall
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm


Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
F-Measure
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm


Baseline
SOCOM++
Baseline Mean
SOCOM++ Mean
F-Measure
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
8
Matching Algorithm