Open-Source Linguistic Techniques for Knowledge Discovery Processing

Khaled Abdalgader¹, Mohamed S. Hajji²

Abstract—Open-source based development provides, in comparison with traditional development methods, foundation code which developers can experiment with and modify according to their experimental needs. This is important in knowledge discovery processing, since it could help researchers to cope with the enormous flexibility of natural language which enables humans to write similar meanings using quite different language structures. In addition to describing computational linguistic techniques relevant to knowledge discovery processing, this paper introduces how the work was developed as open source modules so it can be used for experimental purposes. The open-source developed techniques for knowledge discovery processing include word sense disambiguation step, short-text similarity measurement step, and text clustering step. The concluding remarks discuss these techniques and the benefits of being open-source and show that they are not independent, as the performance of one often depends on the performance of another.

Index Terms—Knowledge Discovery, Short-Text Similarity Measurement, Text Clustering, Open Source Code.

XIX. INTRODUCTION

The availability of large text collections stored in electronic repositories has created the potential of a vast amount of valuable knowledge buried in those texts. This in turn has created the need for automated techniques for deriving relevant and useful knowledge. Knowledge discovery (i.e., text mining) can broadly be described as the process of deriving high quality information from text, where ‘high quality’ refers to some combination of relevance, novelty, and interestingness [1].

Tasks performed in knowledge discovery applications include, typically, text categorization (i.e., classifying a text fragment as belonging to one or more predefined classes or categories) [2]; text clustering (i.e., grouping text fragments according to their degree of similarity to another) [3, 4], and text summarisation (i.e., producing a document summary which captures the main body of relevant content in some document or documents) [5, 6]. These tasks are not independent, an activity focused on text summarisation, for example, may involve sub-tasks involving classification or clustering [7].

Various authors have argued that using open-source tools for knowledge discovery processing helps to avoid problems of inflexibility and to modify functions to fit some needs of the experimental development process. Additionally, the problem of incompatible formats that appear in traditional tools can be avoided. This is important in knowledge discovery processing because of the need to handle the flexibility of natural language where similar meaning can be expressed in a wide range of language structures [8].

Consequently, the availability of open-source subsystems might help researchers to concentrate on experimenting and/or developing some tasks of the discovery process while using ready open-source for other tasks or even for the same task they are working on. Since, while conducting this work, no satisfactory open-source code was found, it was decided to build different parts of the system as open-source components.

This paper introduces the founded computational linguistic techniques relevant to knowledge discovery processing. These include word sense disambiguation, which usually operate at the word level to identify the actual meaning of the word [9-11]; text similarity measurement, where it is necessary to calculate the similarity value between a short-text pair [12-15]; and text clustering, where we are interested in the grouping text fragments according to their degree of similarity [16, 17]. The work was based on developing separate components of open-source code each of which support one of the techniques mentioned above. The availability of such open-source components, or more precisely the lack of it, is discussed too.

The remainder of the paper is structured as follows. Section II starts with a brief introduction to the levels of linguistic processing. Section III presents the developed open-source algorithm to word sense disambiguation (WSD). Section IV introduces the developed open-source sentence-level text similarity method. Developed open-source clustering algorithm that can be used to cluster short-text is presented in Section V. Section VI concludes the paper by presenting a detailed discussion of the developed computational linguistic techniques and the benefits of being open-source and show that they are not independent, as the performance of one often depends on the performance of another.

XX. LEVELS OF LINGUISTIC PROCESSING

As mentioned in the previous section, knowledge discovery processing include text categorization, text clustering and text summarisation. These tasks depend on each other. Consider, for example, the problem of document summarisation [18, 19]. One approach to document summarisation is to identify the main themes or topics which characterise a document, and to then construct a summary of the document by appending, in a coherent
manner, a description of each of those themes. Presumably, fragments of text that are similar to each other are more likely to relate to the same theme than fragments that are less similar. Thus, clustering, using both an appropriate similarity measure and an appropriate level of text fragmentation should provide a useful tool in allowing us to identify those themes.

An important question, then, is what unit of text should be used for clustering on tasks such as text summarisation: words, phrases, sentences, paragraphs, etc.? If the unit of fragmentation is too small (e.g., individual words), we may succeed in finding clusters of related words, but it will be difficult to recombine these words to create a summary. On the other hand, if the unit is too large (e.g., a paragraph or document), then we may not be able to clearly identify themes, since a paragraph may span a number of topics. Sentences are probably at about the right level of fragmentation since they tend to contain information about specific events, and are therefore more likely to provide a suitable context for identifying themes [20, 21].

A second important question concerns representation; i.e., how should sentence-level text be represented in order that an appropriate similarity measure can be defined? Representations such as the Vector Space Model (VSM) [22], which are based solely on word co-occurrence and commonly used at the document level, are clearly not suitable at the sentence level, since two sentences may be about a similar topic, yet contain no words in common. Thus, at the sentence-level, we require a representation which is better able to capture the semantic content of sentences, thereby enabling a more appropriate similarity measure to be defined.

XXI. WORD SENSE DISAMBIGUATION

Word sense disambiguation (WSD)—the process of identifying the appropriate meanings of words as they occur in a sentence—is an intermediate and fundamental task in many natural language processing applications. In this section, we first review the related approaches to WSD and then briefly introduce our developed open-source WSD algorithm.

A. Word Sense Disambiguation Methods

Various WSD methods have been proposed in the literature [9-11], and can be broadly classified as belonging to one of two families: corpus-based approaches, and knowledge-based approaches.

Corpus-based methods utilise supervised learning techniques to induce a classifier from a corpus of training data consisting of a set of labeled words, in which the label indicates the sense in which the word is being used. Once a classifier has been created by extracting the syntactic and semantic features, it can then be used to predict the sense of the target word in novel sentences. In contrast, knowledge-based methods are usually unsupervised and do not require any such corpora, relying instead on external lexical resources such as dictionaries or thesauri [23].

Knowledge-based WSD methods fall into two main groups: similarity-based methods, and graph-based methods. Similarity-based methods determine the sense of a polysemous word by computing the similarity between each of its possible senses and the words in the surrounding context. The correct sense of the target word is then assumed to be that for which the similarity is greatest. Graph-based methods, however, usually build a semantic structure (i.e., a graph) representing all available senses of all of the words being disambiguated. The nodes in this graph correspond to these senses and the edges represent the lexical relations between them. Graph centrality methods are then typically used to determine which nodes are more important (i.e., central) within the graph, and these are considered to be the correct senses of the target words. Because they disambiguate all words in a text fragment simultaneously by exploiting semantic similarities across word senses, graph-based methods usually achieve higher performance than similarity-based methods, which disambiguate words individually, usually without considering the senses assigned to surrounding words [9, 10]. However, the main disadvantage of graphical methods is their high computational complexity.

As a one of the developed open-source linguistic technique introduced in this paper, Abdalghader and Skabar (2012) [9] proposed a new unsupervised similarity-based WSD algorithm that operates by computing the semantic similarity between glosses of the target word and a context vector. The sense of the target word is determined as that for which the similarity between gloss and context vector is highest. This WSD algorithm has previously been reported in [9].

B. Evaluation of Word Sense Disambiguation Methods

WSD methods are often evaluated using a different models of evaluation (i.e., in vitro and in vivo) [9][24] and for in vitro model, there is a several standard datasets have been constructed specifically for this purpose. For example, the Senseval/SeMeVAL campaign provides a shared task with a variety of datasets and sense inventories for all-words and lexical sample settings in different languages. The SemCor [25], Senseval-2 [26] and Senseval-3 [27] datasets are the most common standard datasets.

XXII. SHORT-TEXT SIMILARITY MEASUREMENT

The vector space model has been successful in IR because it is able to adequately capture much of the semantic content of document-level text. This is because documents that are semantically related are likely to contain many words in common, and thus are found to be similar according to popular vector space measures, which are based on word co-occurrence [28]. However, while the assumption that (semantic) similarity can be measured in terms of word co-occurrence may be valid at the document level, the assumption does not hold for small-sized text fragments such as sentences, since two sentences may be semantically related despite having few, if any, words in common.

Various linguistic measures for sentence similarity have been proposed in recent years [12-15]. Rather than representing sentences in a common vector space, most of these measures represent the sentences in a reduced vector space consisting only of the words contained in the

---

3 A series of international WSD competitions (http://www.senseval.org), organized by the ACL-SIGLEX that has been held every three years since 1998.
sentences, and define similarity as some function of inter-
sentence word-to-word similarities, where these similarities
are in turn usually derived either from informational
from some corpora (corpus-based measures), or
semantic information represented in external lexical sources
(knowledge-based measures).

Unlike existing measures, which use the set of exact
words that appear in the sentences, Abdalgader and Skabar
(2011) [12] proposed a novel method (which is one of the
developed open-source linguistic technique described in this
paper) constructs an expansion word set for each sentence
using synonyms of the disambiguated words in that
sentence. This open-source sentence similarity measure has
previously been reported in [12].

A. Evaluation Short-Text Similarity Measures

Sentence similarity measures are commonly evaluated as
in vitro task. To this end, a variety of sentences datasets
have been constructed, and include the Microsoft Research
Paraphrase (MSRP) Corpus [29], and the Recognising
Textual Entailment (RTE) challenge dataset [30]. Each of
these datasets is structured as a collection of sentence pairs,
where each pair is tagged with a binary class value of 1 or 0,
indicating whether the sentences are similar or dissimilar.
The problem is thus a binary classification task in which the
objective is to correctly predict the class membership of the
sentence pair, with performance measured using standard
binary classification measures such as accuracy, precision,
recall and F-measure [31].

One of the difficulties in measuring the performance on in
vitro tasks such as these is that a classification threshold
must be determined, and the performance can be very
sensitive to choice of this threshold. Most researchers
simply use a threshold of 0.5, although some authors have experimented with other threshold values [15].

The need to determine a threshold can also be avoided by using in vivo evaluation model. For example, the task of
clustering relies on a measure of similarity between the
objects (sentences) being clustered and the quality of
clustering can be taken as being indicative of the sentence
similarity measure being used.

XXIII. SHORT-TEXT CLUSTERING

Short-text clustering plays an important role in many text
mining activities. For example consider web mining [32],
where the specific objective might be to discover some
novel information from a set of documents initially retrieved
in response to some query. By clustering the sentences of
those documents we would intuitively expect at least one of
the clusters to be closely related to the concepts described
by the query terms; however, other clusters may contain
information pertaining to the query in some way hitherto
unknown to us, and in such a case we would have
successfully mined new information.

Irrespective of the specific task (e.g., summarisation, text
mining, etc.), most documents will contain interrelated
topics or themes, and many sentences will be related to
some degree to a number of these. This means that ideally a
clustering algorithm should be able to identify soft or fuzzy
clusters [16], in which sentences belong to all of these
clusters with different degrees of membership. However,
problems to which clustering, classification, and other knowledge discovery processing can be applied at the sentence level. However, the performance of any clustering or classification algorithm will be limited by the quality of the input data, and in the case of sentence-level text, performance will depend fundamentally on the quality of the sentence similarity measure that is used.

![Diagram](image)

**Fig. 2** A relationship between developed open-source computational linguistic techniques for knowledge discovery processing.

Different literatures, describe open-source code for different tasks of knowledge discovery processing. To the best of our knowledge, however, there is no proper open source code relevant to this field of research. For example in BabelNet publications like [34], we did not find any mentioning of open-source work. On the other hand, BabelNet provides API tools for multilingual lexical semantic analysis [35]. This tool, though it is helpful, is far from the open source components advocated in this work.

According to developers, MEAD [36] can perform summarization to multiple documents in different languages. It is a public open-source environment used by several implementers for some web applications as well as mobile apps. MEAD is, nevertheless, more concerned with document level processing and does not satisfy this research interests. It is not meant to fit sentence level or to cover related techniques. MEAD framework cannot be scaled down to work with the level of our concerns.

The open-source developed in this work was written in Python with Natural Language Toolkit, the commonly used programming languages in the field of this research, let alone it is becoming very popular in general. The code covers the three main tasks that support text knowledge discovery. It is not completely finalised but was ready for colleagues in our faculty to experiment with. Some colleagues used and modified the code to implement their algorithms concerning different areas of knowledge discovery like similarity measurements and clustering and to check the performance of their ideas. This open-source code proved very helpful and time saving let alone it provided flexibility to researcher. The work of colleagues was a chance also to enhance the open-source code and consolidate its role. Figure 3 shows a portion of our developed open-source linguistic techniques for knowledge discovery processing as implemented in Python environment.

![Diagram](image)

**Fig. 3** A portion of the developed open-source linguistic techniques for knowledge discovery processing as implemented in Python.

In regard to word sense disambiguation, however, it has been shown that graph-based methods usually achieve higher performance than their similarity-based alternatives. This is because they disambiguate all words in a text fragment simultaneously, whereas similarity-based methods disambiguate words individually, usually without considering the senses assigned to surrounding words. The main disadvantage of graphical methods is their high computational complexity. Although similarity-based methods are usually far more efficient than graph-based methods, they are usually based either on gloss-context word overlap [11] or measuring pairwise similarity between word-senses [37], and do not fully utilise the semantic information associated with word-senses that is available through resources such as WordNet's glosses [38]. Also, due to computational requirements, they are usually limited to using context from only a small window surrounding the target word.

Most of the proposed sentence similarity measures compute similarity between their constituent words based either on distributional information from some corpora (corpus-based measures), or on semantic information represented in external sources such as WordNet [38] (knowledge-based measures). Some of the measures, including those of Li et al. (2006) [14] and Mihalcea et al. (2006)[15], incorporate a measure based on the word’s importance, as measured, for example, by the word’s IDF score, or information content, in the case of Li et al. (2006)[14]. The rationale for this is that words which have a higher IDF are more important, and thus should contribute more heavily in the sentence similarity calculation than less important words. However, while it is widely accepted that incorporating IDF scores leads to improved measurement of text similarity at the document level, it is not clear that it has the same utility at the sentence level. Another difficulty in using IDF scores at the sentence level is that many words are polysemous (i.e., have multiple meanings). Even if the sense of
these words can be determined, there remains the problem that IDF scores are generally not available for specific senses of words. Skabar and Abdalgader, 2011 [13] tackles these issue by exploring the idea of incorporating into sentence similarity methods a factor based on the importance of words in the actual sentences being compared (as opposed to average importance over some large corpus).

While a number of papers have reported on the development and evaluation of sentence similarity measures [14, 15, 39], most of these are based on word-to-word similarity using the first sense of each of the words being compared, and intuitively, identifying the correct sense in which a word is being used should lead to a more accurate measure of the similarity between two sentences. To date there has been very little research reporting the incorporation of WSD into sentence similarity measurement. The exceptions are Abdalgader and Skabar (2011) [2] and Ho et al (2010) [40].

In examining the literature on clustering, we have focused on fuzzy relational clustering. This was motivated by the fact that: (i) widely used sentence similarity measures focus on fuzzy relational clustering. This was motivated by the fact that: (i) widely used sentence similarity measures do not represent sentences in a common metric space, thereby requiring a relational, as opposed to attribute-based, approach to clustering, and (ii) sentences are unlikely to relate to just a single concept or theme within a document, but to a number of themes. There are currently very few algorithms falling into this category. Skabar and Abdalgader (2013) [16] presented a new fuzzy clustering algorithm which operates on relational input data; that is, data in the form of a square matrix of pairwise similarities between sentences.

Open-source linguistic techniques are an exciting area of research within the knowledge discovery community, and this paper has introduced several of those techniques which are able to identify word senses, sentence similarity and fuzzy clusters based on relational input data. We have already mentioned some of the new work we are conducting in this area; however, what we are most excited about is extending the techniques to perform opinion mining. The concepts present in natural language documents usually display some type of people opinion, whereas the techniques we have presented in this paper identifies only themes or topics. Our main future objective is to extend these ideas to the development of complete open-source linguistic techniques for sentiment analysis or opinion mining.

REFERENCES
[14] Li, Y., McLean, D., Bandar, Z., O’Shea, F. and Crockett, K. 2006. Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering (TKDE), 18, 8, 1138-1150.


[36] Dragomir Radev; Timothy Allison; Sasha Blair-Goldensohn; John Blitzer; Arda Celebi; Stanko Dimitrov; Elliott Drabek; Ali Hakim; Wai Lam; Danyu Liu; Jahna Otterbacher; Hong Qi; Horacio Saggion; Simone Tesfai; Michael Topper; Adam Winkel; Zha Zhang. MEAD - A Platform for Multidocument Multilingual Text Summarization. *Proceedings of the 4th International Conference on Language Resources and Evaluation* (LREC 2004), Lisbon, Portugal.


