Word Sense Identification Improves the Measurement of Short-Text Similarity

Khaled Abdalgader
Faculty of Computing and Information Technology, Sohar University
P.O. Box: 44, P. Code 311, Sohar, Sultanate of Oman
komar@soharuni.edu.om

ABSTRACT

While a number of short-text similarity measures have recently been proposed, these methods have rarely focused on word sense identification. This paper present a variation, computationally efficient word sense identification method that operates by comparing WordNet glosses of a target word with a context vector comprising the remaining words in the text fragment surrounding the target word. Empirical results show that the method performs favorably against baselines on a two-way sense-assigned dataset, and that its incorporation as a pre-processing step in a sentence similarity measure leads to superior performance on a well-known paraphrase detection dataset.

KEYWORDS

WSI, Short-text similarity and WordNet.

1 INTRODUCTION

Measuring the similarity between small-sized text fragments (e.g., sentences) is a fundamental function in many textual applications. These include text mining and text summarization, which usually operate at the sentence or sub-sentence level [1], [2]; question answering, where it is necessary to calculate the similarity between a question-answer pair [3], [4] and image retrieval, where we are interested in the similarity between a query and an image caption [5].

Various linguistic measures for short text similarity have been proposed in recent years Abdalgader and Skabar (2011) [6], Li *et al*., (2006) [7], Mihalcea *et al*., (2006) [8], Metzler *et al*., (2007) [9], Islam and Inkpen, (2008) [10], Ramage *et al*., (2009) [11], Achananuparp *et al*., (2009) [12], and most of these define the similarity of two text fragments as being some function of the semantic similarities between their constituent words. However, many words have more than one meaning (polysemy), and in order to accurately calculate the similarity between two text fragments, it is therefore important to correctly identify the sense in which the words are being used in those fragments. This sense needs to be determined in the context of the text fragment in which the polysemous word appears [6], and this presents a difficulty, since short text provides only a very limited context.

This paper presents variation of a knowledge-based word sense identification algorithm that identifies the context of a target polysemous word by computing the semantic similarity between WordNet [13] glosses of the target word, and a context vector comprising the remaining words in the text fragment. This is different from current approaches, which, due to computational requirements, are limited to using context from only a small window surrounding the target word. We present empirical results to show that the algorithm compares favorably with baselines on a benchmark word

sense disambiguation (wsd) dataset, and that, when incorporated as a pre-processing step into a sentence similarity measure, results in paraphrase detection performance which surpasses that of other reported methods.

The remainder of the paper is structured as follows. Section 2 provides background into linguistic text similarity measures and word sense identification methods. Section 3 presents the proposed word sense identification method. Section 4 provides a walk-through example, and demonstrates how the method can be incorporated into a sentence similarity measure. Empirical results are presented in Section 5, and Section 6 concludes the paper.

2 BACKGROUND

The vector space model [14] has been successful in IR because it is able to adequately capture much of the semantic content of large documents. This is because large documents may contain many words in common, and thus be found to be similar according to common vector space similarity measures such as the Cosine, Dice or Jaccard measures. However, in the case of sentence-level text, the characteristic flexibility of natural language that enables humans to express similar meanings using quite different sentences in terms of structure and length [15] means that two sentences may be semantically related while containing no words in common. Consequently, a number of sentence similarity measures have recently been proposed, including those by Li *et al*. (2006) [7] and Mihalcea *et al*. (2006) [8].

These sentence similarity measures have two important features in common: (i) rather than representing sentences using the full set of features from some corpora, only the words appearing in the two sentences are used, thus overcoming the problem of data sparseness arising from a full bag-of-words representation, and (ii) they use semantic information derived from external sources to overcome the problem of lack of word co-occurrence.

As an example, consider the approach of Mihalcea *et al*. (2006) [8], who compute similarity between two sentences s_1 and s_2 according to

$$sim_{sem,IDF}(s_1,s_2)=\frac{1}{2}f(T_1,T_2)+\frac{1}{2}f(T_2,T_1) \quad (1)$$

$$f(T_a,T_b)=\sum_{w\in\{s_1\}}\left(\max Sim(w,s_2)\times idf(w)\right)\Big/\sum_{w\in\{s_1\}}idf(w) \quad (2)$$

The computation begins by calculating the similarity score between the first word in s_2 and each word in s_1 that belongs to the same part of speech class. The maximum of these scores is then weighted with the *idf* score of the word from s_2. This procedure is then repeated for the remaining words in s_2, with the weighted maximum scores summed, and then normalized by dividing by the sum of *idf* scores for words in s_1. This procedure is then repeated for s_1. The overall similarity is defined as the average of normalized weighted maximums for s_1 and s_2.

Sentence similarity measures such as those of Mihalcea *et al*. (2006) [7] and Li *et al*. (2006) [8] (the latter of which we use in this research, and describe in Section 3.1), depend in some way on a measure of semantic similarity between words. A number of such measures have been proposed, and we outline some of these in Section 3.2. However, irrespective of the measure used, both of these methods will fail to accurately measure similarity between text-fragments which use different, but synonymous, words to convey the particular meaning. This is because they do not take into account the identification of the actual sense for each word, which intuitively should play an important role in measuring the similarity between text pairs. Incorporation of word sense disambiguation into the sentence similarity computation would be expected, there-

fore, to improve the accuracy of the measure.

2.1 Word Sense Identification

One of the first attempts to automate word sense disambiguation was by Lesk (1986) [16], who employed a knowledge-based dictionary lookup approach. The method determines the sense of a polysemous word by calculating the word overlap between the glosses (i.e., definitions) of two or more target words. The actual senses of the target words are assumed to be those whose glosses have the greatest word overlap. For example, in the case of two words w_1 and w_2, the Lesk score is defined as $\text{Score}_{Lesk}(S_1, S_2) = |gloss(S_1) \cap gloss(S_2)|$, where $S_1 \in Senses(w_1)$, $S_2 \in Senses(w_2)$ and $gloss(S_i)$ is the bag of words in the dictionary definition of sense S_i of w_i. The senses which score the highest value from the above calculation are assigned to the respective words.

While the Lesk algorithm is feasible when the context is small (e.g., two words) it leads to combinatorial explosion as the number of words increases. For example, in a two-word context the number of gloss overlap calculations is $|senses(w_1)| \cdot |senses(w_2)|$, whereas in the case of an n-word context, this increases exponentially to $|senses(w_1)| \cdot |senses(w_2)| \cdot \ldots \cdot |senses(w_n)|$. For this reason, a simplified version of this approach is commonly used, in which the actual sense for word w is selected as the one whose gloss has the greatest overlap with the words in the context of w. That is, $\text{Score}_{LeskVar}(S) = |context(w) \cap gloss(S)|$, where $context(w)$ is the bag of words in a context window that surrounds the word w.

The WordNet hierarchy has been used in a variety of ways to improve on the basic gloss overlap method. For example, rather than just considering glosses of the surrounding words, Banerjee and Pedersen (2002) [17] use the WordNet hierarchy to allow for glosses of senses related to the words in the context to be compared as well; i.e., glosses of surrounding words in the text fragment are expanded to include glosses of those words to which they are related in WordNet. In contrast, Patwardhan *et al*. (2003) [18], following Resnik (1995) [19], take the view that gloss overlap is just another measure of semantic relatedness, and evaluate the use of several word-to-word semantic measures in place of gloss overlap. Due to computational complexity, both [17] and [18] limit the total size of the context window to three or four words.

The method proposed by Sinha and Mihalcea (2007) [20] also uses semantic relatedness, but in this case using a graphical representation of the possible word sense combinations. Graphs consist of a vertex for each sense of each word in the text fragment, with edge weights representing the semantic similarity between the word and sense associated with the nodes connected by that edge. One of four graph centrality algorithms is then used to determine the importance of vertices in the graph, assigning each vertex a score which is then used to identify the most probable sense for each word. In comparison to the approaches of Banerjee and Pedersen (2002) [17] and Patwardhan *et al*. (2003) [18], in which words are disambiguated one at a time using a small context window, Sinha and Mihalcea's (2007) [20] approach disambiguates all words simultaneously. This means that it is likely to result in a more coherent set of meanings over all words in the text fragment. However, the approach is computationally intensive, and requires controlling the complexity by only linking words that appear within a fixed distance of six from each other in the text fragment.

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

3 PROPOSED WSI ALGORITHM

In line with the Lesk algorithm variant described above, the method proposed in this section disambiguates words one at a time; however, rather than using the context provided only in some fixed-size context window surrounding the target word, the method disambiguates the target word using the context provided by all remaining words in the text fragment. Essentially, the algorithm computes the semantic similarity (not overlap) between WordNet glosses of the target polysemous word and the text made up of all of the remaining words in the text fragment, which we refer to as the *context vector*. The target word is then assigned the sense associated with the gloss which has the highest semantic similarity score to the context vector. This procedure is then repeated for all words in the text-fragment.

Note that this method itself relies upon a measure of text similarity, since a gloss (which is a text fragment) must be compared with a context vector (another text fragment). The situation is thus somewhat circular, as our motivation for introducing word sense identification was to improve the measurement of short-text similarity. Attempting to identify the sense of polysemous words in the gloss and context vectors would lead to an infinite regress. Thus, we only perform identification at the top level; i.e., only to polysemous words in the original text fragments s_1 and s_2. We now describe the word sense identification algorithm in detail.

The text fragment containing the words to be identified is first represented as the set of non stop words that it contains:

$$W = \{w_i \mid i=1..N\},$$

where N is the number of words in W. Stop words are removed because they do not carry any semantic information. Suppose that we wish to determine the sense for word w_i from W. Let G_{w_i} be the set of WordNet glosses corresponding to word w_i,; i.e.,

$$G_{w_i} = \left\{ g_{w_i}^k \mid k = 1..N_{w_i} \right\}$$

where N_{w_i} is the number of WordNet senses for w_i, and $g_{w_i}^k$ is the set of non stop words in the k^{th} WordNet gloss of w_i. Let R_i be the context vector comprising all words from W, except w_i:

$$R_i = \left\{ w_j \mid w_j \in W \text{ and } j \neq i \right\}$$

The sense for word w_i is identified as the k value for which $g_{w_i}^k$ is semantically most similar to R_i. The full procedure is described in Algorithm 1.

Algorithm 1: Word Sense Identification

Input: Words $W = \{w_i \mid i = 1..N\}$
Glosses $G_{w_i} = \left\{ g_{w_i}^k \mid k = 1..N_{w_i} \right\}, i = 1..N$

Output: WordNet senses $T = \{t_i \mid i = 1..N\}$ where t_i is the WordNet sense of w_i

Word Sense Identification

1: **for** i = 1 to N **do**
2: $R_i = \left\{ w_j \mid w_j \in W \text{ and } j \neq i \right\}$
3: max_sim ← 0
4: $t_i \leftarrow 1$
5: **for** j = 1 to N_{w_i} **do**
6:

$$\text{tmp} \leftarrow \text{similarity}\left(\text{morphy}\left(g_{w_i}^j\right), \text{morphy}\left(R_i\right)\right)$$

7: **if** tmp > max_sim **then**
8: max_sim ← tmp
9: $t_i \leftarrow j$
10: **end if**
11: **end for**
12: **end for**

Note that in line 6, in which similarity is calculated, that a function 'morphy' has been applied to both the gloss and context vectors before the similarity is calculated. This function takes a set of words as input, and returns a set of the same length consist-

ing of the morphological stems of the original words.

3.1 Gloss-Context Vectors Similarity

To calculate the similarity between gloss and context vectors (line 6 of algorithm), we use a variation on the method proposed by Li *et al*. (2006) [7]. Let W_1 and W_2 be the word sets of the two text fragments whose similarity we wish to calculate. Assume that W_1 is the gloss vector corresponding to the WordNet sense k for the target word w_i and W_2 is the corresponding context vector:

$$W_1 = g_{w_i}^k = \{w_{1i} \mid i = 1..N_1\}$$

$$W_2 = R_i = \{w_{2i} \mid i = 1..N_2\}$$

We first form the union word set U by combining all distinct words from W_1 and W_2:

$$U = W_1 \cup W_2 = \{w_i \mid i = 1..N \le N_1 + N_2\}$$

where N is the total number of words in U. The union word set can be viewed as the semantic information board for the compared fragments. We now construct semantic vectors $\mathbf{V}_1$ and $\mathbf{V}_2$, corresponding to W_1 and W_2 respectively. Each entry of these vectors corresponds to a word in the union set U, so their dimension is N. Let v_{ij} be the j^{th} element of $\mathbf{V}_i$, and let w_j be the corresponding word from U. The value of v_{ij} is determined according to the semantic similarity of w_j to all words in W_i. There are two cases to consider, depending on whether w_j appears or does not appear in W_i :

Case 1: w_j appears in W_i.
Set v_{ij} equal to 1.

Case 2: w_j does not appear in W_i.
Calculate a semantic similarity score between w_j and each word in W_i. Assign the highest of these scores to v_{ij}.

The semantic similarity score used in Case 2 is the *shortest path* measure. This measure is capable of calculating the similarity between words with different part-of-speech. This reason, together with the computational simplicity of the approach, makes it appropriate to use within this algorithm. Once $\mathbf{V}_1$ and $\mathbf{V}_2$ have been determined, the semantic similarity is measured as the Cosine similarity between $\mathbf{V}_1$ and $\mathbf{V}_2$: $\text{similarity}(W_1, W_2) = (\mathbf{V}_1 \cdot \mathbf{V}_2) / (|\mathbf{V}_1||\mathbf{V}_2|)$.

3.2 Word-to-Word Semantic Similarity

The method described above relies on a semantic word similarity measure. A number of such measures have been defined. In the following we describe the six measures we use in this paper, all of which are based on WordNet. A comprehensive review can be found in [21].

The shortest path similarity [22] is the simplest measure we use, and is defined as:

$$Sim_{Path}(w_1, w_2) = \frac{1}{length(w_1, w_2)} \quad (3)$$

where *length* is the length of the shortest path between two words using node-counting (including the end nodes) in the WordNet hierarchy.

The shortest path measure does not take into account the depth of words in the hierarchy, and since words at top levels have more general semantics and less similarity between them than words at lower levels, it is widely believed that better measures can be defined by taking depth into account. Leacock and Chodorow (1998) [23] define similarity as:

$$Sim_{lch}(w_1, w_2) = -\log \frac{N_p}{2D} \quad (4)$$

where N_p is the distance between the words and D is the maximum depth in the hierarchy.

The measure proposed by Wu and Palmer (1994) [24] computes the semantic similarity of the two words as a func-

tion of the path length from the least common subsumer (LCS); i.e., the words' deepest common ancestor in the hierarchy:

$$Sim_{Lch}(w_1, w_2) = \frac{2 * depth\left(LCS(w_1, w_2)\right)}{depth(w_1) + depth(w_2)} \quad (5)$$

where depth(w) is the depth of word w.

The Resnik (1995) [19] measure is based on the idea that the degree to which two words are similar is proportional to the amount of information they share. The measure is defined as the information content (IC) of the LCS of the two words:

$$Sim_{res}(w_1, w_2) = IC(LCS(w_1, w_2)) \quad (6)$$

where $IC(w)$ is defined as $IC(w) = -\log P(w)$, where $P(w)$ is the probability that word w appears in a large corpus.

The Lin (1998) [25] measure normalizes the Resnik measure by dividing it by the average information content of w_1 and w_2:

$$Sim_{Lin}(w_1, w_2) = \frac{2 \times IC(LCS(w_1, w_2))}{IC(w_1) + IC(w_2)} \quad (7)$$

The measure proposed by Jiang and Conrath (1997) [26] is also based on information content and is defined as:

$$Sim_{J\&C}(w_1, w_2) = \frac{1}{IC(w_1) + IC(w_2) - 2 \times IC(LCS(w_1, w_2))} \quad (8)$$

3.3 Computational Complexity

The word sense identification algorithm utilizes the context provided by all words in the text-fragment, and not just those within some fixed-size window surrounding the target word. Importantly, this does not lead to significantly increased computational requirements.

To demonstrate, it is useful to compare the computational complexity of our approach with that of the context window approach used in [17] and [18]. Suppose that the average number of WordNet senses per word is a, the average length of text fragments and WordNet glosses (after removal of stopwords) is N, and in the case of a context-window approach, that the total number of words used in the context window n. Since both approaches disambiguate words one at a time, we consider the cost of disambiguating a single word. The number of word-to-word similarity calculations required to disambiguate a word using the context window approach is quadratic in a (since all senses of the target word must be compared with all sense of the neighboring words), and linear in n. In our approach, the number of similarity calculations is linear in a (since only the various senses of the target word are considered, with first sense used for words in the context vector), but is quadratic in N (since determining the semantic vectors requires calculating the similarity between words in the union set with words appearing in the original sentences). However, at the sentence level the values of N and a will be roughly comparable, and thus the computational requirements of the proposed approach will usually not be much greater than those of the context window approach. If computation requirements are prohibitive, it is of course possible to also adopt a windowing approach within our method, using the context provided by all words within this window.

4 A WALK-THROUGH EXAMPLE

For clarity, we now provide an example of the method described in the previous section. Consider the following two sentences, which contain the polysemous words 'virus' and 'bank':

S_1: *The virus spread in all saving deposit money systems in the bank.*

S_2: *All fish in the river of the south bank have been infected by the virus.*

Firstly, we show how the word sense identification algorithm identifies the sense for each word in S_1. Suppose that

we wish to identify the sense of word 'virus'. We construct the set of non stop words appearing in the sentence S_1:

W = {'virus', 'spread', 'saving', 'deposit', 'money', 'systems', 'bank'}

The word 'virus' has three WordNet glosses:

Sense 0: (virology) ultramicroscopic infectious agent that replicates itself only within cells of …

Sense 1: a harmful or corrupting agency.

Sense 2: a software program capable of reproducing itself and usually capable of causing …

We then construct the sets g^0_{virus}, g^1_{virus} and g^2_{virus} corresponding to each of these glosses. For space reasons, we show only the second: g^1_{virus} = {'harmful', 'corrupt', 'agency'}

Since we are identifying the sense of 'virus', the context vector R_{virus} will consist of all words from W except 'virus':

R_{virus} = {'spread', 'saving', 'deposit', 'money', 'system', 'bank'}

The union set is formed by combining words from g^1_{virus} and R_{virus}:

U = {'saving', 'money', 'agency', 'system', 'harmful', 'spread', 'deposit', 'corrupt', 'bank'}

Note that in constructing the union set, morphological stemming has been applied to the words from W_1 and W_2, as per line 6 of the word sense identification algorithm, but has not been applied to the target polysemous word.

We now calculate the semantic vectors $\mathbf{V}_1$ and $\mathbf{V}_2$, corresponding to g^1_{virus} and R_{virus} respectively. resulting in

$\mathbf{V}_1$ = (0.083, 0.083, 1.0, 0.071, 1.0, 0.076, 0.071, 1.0, 0.076)

$\mathbf{V}_2$ = (1.0, 1.0, 0.083, 1.0, 0.0, 1.0, 1.0, 0.0, 1.0)

Finally, calculating the cosine of these vectors gives a similarity score of 0.128.

Using the first and third glosses results in similarity scores of 0.201 and 0.243, and thus the sense of word *virus* in the original sentence will be determined as that corresponding to the third WordNet sense, which is clearly the sense in which *virus* would be interpreted by a human in this context.

Repeating this same procedure for all words in W results in the following set of sense-assigned words:

S_1 = {('virus', 2), ('spread', 18), ('saving', 7), ('deposit', 3), ('money', 2), ('systems', 0), ('bank', 1)}

where the numbers indicate the sense of the associated word. Repeating for the second sentence results in the following:

S_2 = {('fish', 0), ('river', 0), ('south', 3), ('bank', 0), ('infected', 4), ('virus', 0)}

We now use the sense assignments to calculate the semantic similarity score between the original two sentences. The union set of sense-assigned words from S_1 and S_2 is

U = {('river', 0), ('virus', 0), ('spread', 18), ('bank', 0), ('systems', 0), ('virus', 2), ('money', 2), ('saving', 7), ('deposit', 3), ('south', 3), ('fish', 0), ('bank', 1), ('infected', 4)}

and the resulting semantic vectors are

$\mathbf{V}_1$ = (0.113, 0.172, 1.0, 0.152, 1.0, 1.0, 1.0, 1.0, 1.0, 0.156, 0.199, 1.0, 0.0)

$\mathbf{V}_2$ = (1.0, 1.0, 0.0, 1.0, 0.199, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 1.0)

Calculating the cosine of these vectors using the Lin measure results in a sentence similarity score of 0.15. This value is relatively low, given the distribution of values we have observed by comparing many pairs of similar and dissimilar sentences (see Figure 1 for some typical distributions), and indicates that the two sentences are not semantically related, despite containing common words.

5 EMPIRICAL RESULTS

In this section we present results from applying the algorithm to two benchmark datasets: the Two-Way Ambiguities (TWA) dataset [27], and the Microsoft Research Paraphrase Corpus (MSRP) [28] dataset. All results reported are based on the use of the lexical knowledge base WordNet

3.0.

The TWA dataset is a binary sense-tagged dataset for six English words with two-way ambiguities: *bass*, *crane*, *motion*, *palm*, *plant* and *tank*, all of which are nouns. The dataset was explicitly established for word sense identification research purposes. The test data contains 993 instances, which were drawn from the British National Corpus.

Table 1 shows the accuracy, precision, recall, and F-measure resulting from applying the wsd method to the TWA dataset. This dataset has been used mainly for testing supervised methods, and we are not aware of any unsupervised methods which have been applied to it. However, according to [27], the most-frequent-sense baseline (i.e., the performance resulting from simply assigning the most common sense) is a tough baseline for an unsupervised word sense identification method to beat. Given that our method performs very favorably against this baseline (average accuracy of 75.6% versus 70.6% for the baseline), we believe that the ability of the algorithm to identify the correct sense of polysemous words is sufficient for use within a text similarity measure.

Table 1. Wsd performance on TWA test set.

Word	Meaning	Acc	Prec	Rec	F
Bass	Fish / Music	94.3	64.2	90.0	75.0
Crane	Bird / Machine	74.7	48.2	60.8	53.8
Motion	Physical / Legal	72.6	64.2	15.2	24.6
Palm	Hand / Tree	73.6	100	8.6	15.8
Plant	Living / Factory	65.4	58.1	87.2	69.7
Tank	Container / Vehicle	73.1	83.8	34.6	49.0
Accuracy Average		75.6			
Most frequent sense Baseline					
MFS Acc Average		70.6			

The MSRP dataset consists of 5801 pairs of text-fragments that were automatically collected from a large number of web newswire postings over a period of 18 months. Each pair was manually labeled with a binary true or false value by two human annotators, indicating whether or not the two fragments in a pair were considered a paraphrase of each other. The agreement between the human judges was 83%. The corpus is divided into 4076 training pairs and 1725 test pairs. Since the algorithm performs unsupervised word sense identification, we use only the test data.

Table 2 shows the results of applying the sentence similarity measure to identifying paraphrases on the MSRP dataset. The first section of the table shows performance with the use of wsd; the second shows performance without (i.e., word-to-word semantic similarity is based on the first WordNet sense of the component words). In each case we use all six of the word-to-word semantic similarity measures described in Section 3.2.

Table 2. Performance on MSRP data

Measure	Acc	Prec	Rec	F
Similarity Measure *with* Word Sense Identification				
Path	71.5	77.2	80.9	79.0
Wup	68.9	69.2	95.9	80.4
Lch	66.4	66.6	99.4	79.7
Lin	**75.2**	77.5	88.4	**82.6**
Resnik	67.3	67.4	98.4	80.0
J&C	71.6	77.8	80.1	78.9
Similarity Measure *without* Word Sense Identification				
Path	71.7	70.8	97.7	82.1
Wup	66.5	66.5	99.9	79.8
Lch	66.5	66.5	99.9	79.8
Lin	69.4	68.7	99.1	81.1
Resnik	66.4	66.4	100	79.8
J&C	72.0	71.2	97.0	82.1

For comparative purposes, Table 3 shows the performance of some other algorithms that have been reported in the literature, all of which are based on a 0.5 calculation threshold: Mihalcea *et al*.'s (2006) [8] corpus-based and WordNet-based measures; the random graph walk method of Ramage *et al*. (2009) [11] using three distributional similarity measures; and, following Mihalcea *et al*. (2006) [8], a baseline that meas-

ures the cosine similarity between vectors in a full bag-of-words representation with *tf-idf* weighting, and a random baseline, created by randomly assigning a true or false value to pairs of text fragments.

Table 3. MSRP Performance of other approaches

Measure	Acc	Prec	Rec	F
Mihalcea *et al*., Corpus-based				
PMI-IR	69.9	70.2	95.2	81.0
LSA	68.4	69.7	95.2	80.5
Mihalcea *et al*., WordNet-based				
Lin	69.3	71.6	88.7	79.2
J&C	69.3	72.2	87.1	79.0
Resnik	69.0	69.0	96.4	80.4
Ramage *et al.*, Random Graph Walk				
Cosine	68.7	-	-	78.7
Dice	70.8	-	-	80.1
JS	68.8	-	-	80.5
Baselines				
Vector-based	65.4	71.6	79.5	75.3
Random	51.3	68.3	50.0	57.8

There are several observations to make from the data in Tables 2 and 3. Firstly, note in the second section of Table 2 (similarity without wsd) the consistently low and high values respectively for precision and recall. This indicates that the 0.5 classification threshold is too low, resulting in a small number of false negatives (paraphrases classified as non-paraphrases) at the expense of a large number of false positives (non-paraphrases classified as paraphrases). This is verified in Figure 1(a) and 1(c), which show the distribution of similarity values corresponding to the Lch and J&C measures respectively. In both cases the distributions are skewed towards the right. It is also interesting to note from the second section of Table 2 that the accuracy achieved using the Path measure (71.7%) and the J&C measure (72.0%) exceed all of the accuracies shown in Table 3; however, these values are still associated with an unacceptable balance between precision and recall.

Turning now to the top section of Table 2, in which WSI was used, we see that there are three cases in which the accuracy clearly exceeds those quoted in Table 3 (Path, Lin and J&C). Importantly, we note that in each of these cases a much better balance is achieved between precision and recall than was the case without the use of WSI. This is partly a result of the more evenly distributed similarity values, as can be seen for the case of the J&C measure in Figure 1(c) and 1(d), which show the distributions of values with and without wsd respectively. The use of WSI leads to smaller similarity values because the word-to-word similarity between two different senses of the same word will always be less than or equal to the similarity value resulting from comparing a particular sense of a word with itself (i.e., a value of 1). Note, however, that in the case of the Lch measure (Figure 1(b)), the distribution is heavily skewed to the right, even with WSI, resulting in performance no different to the case of not using wsd.

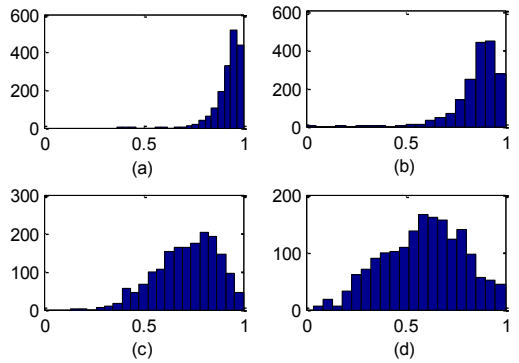

Figure 1. Distribution of similarity scores on MSRP data: (a) Lch with WSI; (b) Lch without WSI; (a) J&C with WSI; (b) J&C without WSI

The classification threshold used in our results, as well as all of the results from Table 3, is 0.5. While some improvement might be obtained by tuning this threshold, this would require use of a training set. We are only interested in unsupervised classification, so we do not perform any such tuning.

Many applications which might require sentence similarity to be measured are in fact not classification problems, and thus do not require thresholding. Sentence clustering is an example. For these tasks, it is

not so important that different measures yield similar values. What is more important is that the values are highly correlated. To this end, Table 4 shows the Pearson correlation between similarity values obtained for the MSRP dataset. It is interesting to note that the methods which result in the most highly correlated values (Path, Lin and J&C) are precisely those which were found to give the best performance when used with WSI in Table 2.

Table 4. MSRP similarity measure correlations

	Path	Wup	Lch	**Lin**	Resnik	**J&C**
Path	1.000	0.874	0.653	**0.971**	0.741	**0.993**
Wup		1.000	0.745	0.927	0.928	0.848
Lch			1.000	0.690	0.819	0.601
Lin				1.000	0.819	**0.962**
Resnik					1.000	0.702
J&C						1.000

6 CONCLUSIONS

A variation of word sense identification method has been presented. The method utilizes the context provided by all words in a text-fragment. Importantly, this does not lead to significantly increased computational requirements over the more conventional approach of using a fixed-size context window. The method shows good performance against baseline measures on the TWA dataset, and, when used as a preprocessing step for sentence similarity calculation, leads to improved paraphrase detection performance on the MSRP dataset.

Our primary interest in this area lies in the context of sentence clustering. Clustering methods rely on a similarity measure, and hence our focus in this paper on short-text similarity measures. We are currently experimenting with these measures on sentence clustering tasks.

The effectiveness of knowledge-based sentence similarity measures such as those used in this paper clearly depends on the quality of the lexical resource. WordNet has been criticized as being limited in regards to its coverage [29], and we are currently exploring the feasibility of using larger knowledge bases such as Wikipedia as either a replacement for, or an adjunct to, WordNet.

7 REFERENCES

[1] Atkinson-Abutridy, John, Chris Mellish, and Stuart Aitken. "Combining Information Extraction with Genetic Algorithms for Text Mining". IEEE Intelligent Systems, 19(3):22-30, 2004.

[2] Erkan, Gunes, and Dragomir R. Radev. "LexRank: Graph-based lexical centrality as salience in text summarization". Journal of Art. Int. Research, 22, 2004, pages 457-479.

[3] Bilotti, Matthew W., Paul Ogilvie, Jamie Callan, and Eric Nyberg. "Structured retrieval for question answering". In Proc. SIGIR 2007, ACM, New York, 2007, pages 351-358.

[4] Mohler, Michael, and Rada Mihalcea. "Text-to-Text semantic similarity for automatic short answer grading". In Proc. EC-ACL 2009, 2009, pages 567-575, Athens, Greece.

[5] Coelho, Tatiana A. S., Pavel P. Calado, Lamarque V. Souza, Berthier Ribeiro-Neto, and Richard Muntz. "Image retrieval using multiple evidence ranking". IEEE Tran. On KDD, 16(4):408-417, 2004.

[6] Khaled Abdalgader and Andrew Skabar. "Short-text similarity measurement using word sense disambiguation and synonym expansion". In Proc. AI 2010 :Advances in Artificial Intelligence, Lecture Notes in Computer Science, Australia, 2011, 6464, pages 435–444.

[7] Li, Yuhua, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. "Sentence similarity based on semantic nets and corpus statistics". IEEE Trans. on Knowledge and Data Engineering, 18(8):1138–1150, 2006.

[8] Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. "Corpus-based and Knowledge-based measures of text semantic similarity". In Proc. of the 21st National Conference on Art. Int., Boston, 1, 2006, pages 775-780.

[9] Metzler, Donald, Susan T. Dumais, and Christopher Meek. "Similarity measures for short segments of text". In Proc. of the 29th European Conference on Information Retrieval, 4425, Springer, Heidelberg, 2007, pages 16-27.

[10] Islam, Aminul, and Diana Inkpen. "Semantic text similarity using corpus-based word similarity and string similarity". ACM Trans. on KDD, 2(2):1-25, 2008.

[11] Ramage, Daniel, Anna N. Rafferty, and Christopher D. Manning. "Random Walks for Text Semantic Similarity". In: Proc. ACL-IJCNLP 2009, 2009, pages 23-31, Suntec, Singapore.

[12] Achananuparp, Palakorn, Xiaohua Hu, and Christopher C. Yang. "Addressing the variability of natural language expression in sentence similarity with semantic structure of the sentences". In Proc. PAKDD 2009, 2009, pages 548-555.

[13] Fellbaum, C. "WordNet: An Electronic Lexical Database". MIT Press, 1998, Cambridge.

[14] Salton, Gerard. "Automatic text processing: the transformation, analysis, and retrieval of information by computer". Addison-Wesley, Reading, Mass, 1989.

[15] Bates, Marcia J. "Subject access in online catalogue: A design model". Journal of the American Society for Information Science, 37(6):357-376, 1986.

[16] Lesk, Michael. "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone". In Proc. of the SIGDOC 1986, 1986, pages 24-26, Toronto, Canada.

[17] Banerjee, Satanjeev, and Ted Pedersen. "An adapted Lesk algorithm for word sense disambiguation using Word-Net". In Proc. ITPCL 2002, 2002, pages 136-145.

[18] Patwardhan, Siddharth, Satanjeev Banerjee, and Ted Pedersen. "Using measures of semantic relatedness for word sense disambiguation". In Proc. ITPCL 2003, 2003, pages 241-257, Mexico City.

[19] Resnik, Philip. "Using information content to evaluate semantic similarity". In Proc. of the 14th International Joint Conf. on Art. Int., 1, 1995, pages 448-453, Montreal.

[20] Sinha, Ravi, and Rada Mihalcea. "Unsupervised Graph-based word sense disambiguation using measures of word semantic similarity". In Proc. IEEE (ICSC 2007), 2007, pages 363-369, Irvine, CA.

[21] Budanitsky, Alexander, and Graeme Hirst. "Evaluating WordNet-based measures of lexical semantic relatedness". Computational Linguistics, 32(1):13-47, 2006.

[22] Rada, Roy, Hafedh Mili, Ellen Bicknell, and Maria Blettner. "Development and application of a metric to semantic nets". IEEE Trans. Sys., Man and Cyb., 19(1): 17–30, 1989.

[23] Leacock, Claudia, and Martin Chodorow. "Combining local context and WordNet similarity for word sense identification". In C. Fellbaum (Ed.), Cambridge, Mass.: MIT Press, 1998, Chp. 11, pages 265-283.

[24] Wu, Zhibiao, and Martha Palmer. "Verb semantics and lexical selection". In Proc. of the 32nd Annual Meeting of the Association for Computational Linguistics, 1994, pages 133-138, Las Cruces, New Mexico.

[25] Lin, Dekang. "An information-theoretic definition of similarity". In Proc. of the 15th International Conf. on Machine Learning, Madison, Wisc, 1998, pages 296-304.

[26] Jiang, Jay J., and David W. Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy". In Proc. of the 10th International Conf. on Research in Computational Linguistics, 1997, pages 19-33, Taipei, Taiwan.

[27] Mihalcea, Rada. "The role of non-ambiguous words in natural language disambiguation". In Proc. of the Conference on Recent Advances in Natural Language Processing, RANLP, 2003, Borovetz, Bulgaria.

[28] Dolan, William, Chris Quirk, and Chris Brockett. "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources". In Proc. of the 20th International Conf. on Computational Linguistics, 2004, pages 350-356.

[29] Achananuparp, Palakorn, Xiaohua Hu, and Xiajiong Shen. "The evaluation of sentence similarity measures". In Proc. DWKD 2008, 5182, Springer, Heidelberg, 2008, pages 305-316.