Measuring the Semantic Similarity of Texts

Courtney Corley and **Rada Mihalcea**
Department of Computer Science
University of North Texas
{corley,rada}@cs.unt.edu

Abstract

This paper presents a knowledge-based method for measuring the semantic-similarity of texts. While there is a large body of previous work focused on finding the semantic similarity of concepts and words, the application of these word-oriented methods to text similarity has not been yet explored. In this paper, we introduce a method that combines word-to-word similarity metrics into a text-to-text metric, and we show that this method outperforms the traditional text similarity metrics based on lexical matching.

1 Introduction

Measures of text similarity have been used for a long time in applications in natural language processing and related areas. One of the earliest applications of text similarity is perhaps the vectorial model in information retrieval, where the document most relevant to an input query is determined by ranking documents in a collection in reversed order of their *similarity* to the given query (Salton and Lesk, 1971). Text similarity has been also used for relevance feedback and text classification (Rocchio, 1971), word sense disambiguation (Lesk, 1986), and more recently for extractive summarization (Salton et al., 1997b), and methods for automatic evaluation of machine translation (Papineni et al., 2002) or text summarization (Lin and Hovy, 2003).

The typical approach to finding the similarity between two text segments is to use a simple lexical matching method, and produce a similarity score based on the number of lexical units that occur in both input segments. Improvements to this simple method have considered stemming, stop-word removal, part-of-speech tagging, longest subsequence matching, as well as various weighting and normalization factors (Salton et al., 1997a). While successful to a certain degree, these lexical matching similarity methods fail to identify the *semantic* similarity of texts. For instance, there is an obvious similarity between the text segments *I own a dog* and *I have an animal*, but most of the current text similarity metrics will fail in identifying any kind of connection between these texts. The only exception to this trend is perhaps the latent semantic analysis (LSA) method (Landauer et al., 1998), which represents an improvement over earlier attempts to use measures of semantic similarity for information retrieval (Voorhees, 1993), (Xu and Croft, 1996). LSA aims to find similar terms in large text collections, and measure similarity between texts by including these additional related words. However, to date LSA has not been used on a large scale, due to the complexity and computational cost associated with the algorithm, and perhaps also due to the "black-box" effect that does not allow for any deep insights into why some terms are selected as similar during the singular value decomposition process.

In this paper, we explore a knowledge-based method for measuring the semantic similarity of texts. While there are several methods previously proposed for finding the semantic similarity of words, to our knowledge the application of these word-oriented methods to text similarity has not been yet explored. We introduce an algorithm

Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pages 13–18,
Ann Arbor, June 2005. ©2005 Association for Computational Linguistics

that combines the word-to-word similarity metrics into a text-to-text semantic similarity metric, and we show that this method outperforms the simpler lexical matching similarity approach, as measured in a paraphrase identification application.

2 Measuring Text Semantic Similarity

Given two input text segments, we want to automatically derive a score that indicates their similarity at *semantic* level, thus going beyond the simple lexical matching methods traditionally used for this task. Although we acknowledge the fact that a comprehensive metric of text semantic similarity should take into account the relations between words, as well as the role played by the various entities involved in the interactions described by each of the two texts, we take a first rough cut at this problem and attempt to model the semantic similarity of texts as a function of the semantic similarity of the component words. We do this by combining metrics of word-to-word similarity and language models into a formula that is a potentially good indicator of the semantic similarity of the two input texts.

2.1 Semantic Similarity of Words

There is a relatively large number of word-to-word similarity metrics that were previously proposed in the literature, ranging from distance-oriented measures computed on semantic networks, to metrics based on models of distributional similarity learned from large text collections. From these, we chose to focus our attention on six different metrics, selected mainly for their observed performance in natural language processing applications, e.g. malapropism detection (Budanitsky and Hirst, 2001) and word sense disambiguation (Patwardhan et al., 2003), and for their relatively high computational efficiency.

We conduct our evaluation using the following word similarity metrics: Leacock & Chodorow, Lesk, Wu & Palmer, Resnik, Lin, and Jiang & Conrath. Note that all these metrics are defined between concepts, rather than words, but they can be easily turned into a word-to-word similarity metric by selecting for any given pair of words those two meanings that lead to the highest concept-to-concept similarity. We use the WordNet-based implementation of these metrics, as available in the WordNet::Similarity package (Patwardhan et al., 2003). We provide below a short description for each of these six metrics.

The **Leacock & Chodorow** (Leacock and Chodorow, 1998) similarity is determined as:

$$Sim_{lch} = -\log \frac{length}{2 * D} \tag{1}$$

where $length$ is the length of the shortest path between two concepts using node-counting, and D is the maximum depth of the taxonomy.

The **Lesk** similarity of two concepts is defined as a function of the overlap between the corresponding definitions, as provided by a dictionary. It is based on an algorithm proposed in (Lesk, 1986) as a solution for word sense disambiguation.

The **Wu and Palmer** (Wu and Palmer, 1994) similarity metric measures the depth of the two concepts in the WordNet taxonomy, and the depth of the least common subsumer (LCS), and combines these figures into a similarity score:

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \tag{2}$$

The measure introduced by **Resnik** (Resnik, 1995) returns the information content (IC) of the LCS of two concepts:

$$Sim_{res} = IC(LCS) \tag{3}$$

where IC is defined as:

$$IC(c) = -\log P(c) \tag{4}$$

and $P(c)$ is the probability of encountering an instance of concept c in a large corpus.

The next measure we use in our experiments is the metric introduced by **Lin** (Lin, 1998), which builds on Resnik's measure of similarity, and adds a normalization factor consisting of the information content of the two input concepts:

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)} \tag{5}$$

Finally, the last similarity metric we consider is **Jiang & Conrath** (Jiang and Conrath, 1997), which returns a score determined by:

$$Sim_{jnc} = \frac{1}{IC(concept_1) + IC(concept_2) - 2 * IC(LCS)} \tag{6}$$

2.2 Language Models

In addition to the semantic similarity of words, we also want to take into account the *specificity* of words, so that we can give a higher weight to a semantic matching identified between two very specific words (e.g. *collie* and *sheepdog*), and give less importance to the similarity score measured between generic concepts (e.g. *go* and *be*). While the specificity of words is already measured to some extent by their depth in the semantic hierarchy, we are reinforcing this factor with a corpus-based measure of word specificity, based on distributional information learned from large corpora.

Language models are frequently used in natural language processing applications to account for the distribution of words in language. While word frequency does not always constitute a good measure of word importance, the distribution of words across an entire collection can be a good indicator of the *specificity* of the words. Terms that occur in a few documents with high frequency contain a greater amount of discriminatory ability, while terms that occur in numerous documents across a collection with a high frequency have inherently less meaning to a document. We determine the *specificity* of a word using the inverse document frequency introduced in (Sparck-Jones, 1972), which is defined as the total number of documents in the corpus, divided by the total number of documents that include that word. In the experiments reported in this paper, we use the British National Corpus to derive the document frequency counts, but other corpora could be used to the same effect.

2.3 Semantic Similarity of Texts

Provided a measure of semantic similarity between words, and an indication of the word specificity, we combine them into a measure of text semantic similarity, by pairing up those words that are found to be most similar to each other, and weighting their similarity with the corresponding specificity score.

We define a *directional* measure of similarity, which indicates the semantic similarity of a text segment T_i *with respect to* a text segment T_j. This definition provides us with the flexibility we need to handle applications where the directional knowledge is useful (e.g. entailment), and at the same time it gives us the means to handle bidirectional similarity through a simple combination of two unidirectional metrics.

For a given pair of text segments, we start by creating *sets* of open-class words, with a separate set created for nouns, verbs, adjectives, and adverbs. In addition, we also create a set for cardinals, since numbers can also play an important role in the understanding of a text. Next, we try to determine pairs of similar words across the sets corresponding to the same open-class in the two text segments. For nouns and verbs, we use a measure of semantic similarity based on WordNet, while for the other word classes we apply lexical matching[1].

For each noun (verb) in the set of nouns (verbs) belonging to one of the text segments, we try to identify the noun (verb) in the other text segment that has the highest semantic similarity ($maxSim$), according to one of the six measures of similarity described in Section 2.1. If this similarity measure results in a score greater than 0, then the word is added to the set of similar words for the corresponding word class WS_{pos}[2]. The remaining word classes: adjectives, adverbs, and cardinals, are checked for lexical similarity with their counter-parts and included in the corresponding word class set if a match is found.

The similarity between the input text segments T_i and T_j is then determined using a scoring function that combines the word-to-word similarities and the word specificity:

$$sim(T_i,T_j)_{T_i} = \frac{\sum\limits_{pos}(\sum\limits_{\mathbf{w}_k \in \{WS_{pos}\}} (maxSim(\mathbf{w}_k) * idf_{\mathbf{w}_k}))}{\sum\limits_{\mathbf{w}_k \in \{T_{i_{pos}}\}} idf_{\mathbf{w}_k}} \tag{7}$$

This score, which has a value between 0 and 1, is a measure of the directional similarity, in this case computed with respect to T_i. The scores from both directions can be combined into a bidirectional similarity using a simple average function:

$$sim(T_i,T_j) = \frac{sim(T_i,T_j)_{T_i} + sim(T_i,T_j)_{T_j}}{2} \tag{8}$$

[1] The reason behind this decision is the fact that most of the semantic similarity measures apply only to nouns and verbs, and there are only one or two relatedness metrics that can be applied to adjectives and adverbs.

[2] All similarity scores have a value between 0 and 1. The similarity threshold can be also set to a value larger than 0, which would result in tighter measures of similarity.

Text Segment 1: The jurors were taken into the courtroom in groups of 40 and asked to fill out a questionnaire.

- Set_{NN} = {juror, courtroom, group, questionnaire}
  Set_{VB} = {be, take, ask, fill}
  Set_{RB} = {out}
  Set_{CD} = {40}

Text Segment 2: About 120 potential jurors were being asked to complete a lengthy questionnaire.

- Set_{NN} = {juror, questionnaire}
  Set_{VB} = {be, ask, complete}
  Set_{JJ} = {potential, lengthy}
  Set_{CD} = {120}

Figure 1: Two text segments and their corresponding word class sets

3 A Walk-Through Example

We illustrate the application of the text similarity measure with an example. Given two text segments, as shown in Figure 1, we want to determine a score that reflects their semantic similarity. For illustration purposes, we restrict our attention to one measure of word-to-word similarity, the **Wu & Palmer** metric.

First, the text segments are tokenized, part-of-speech tagged, and the words are inserted into their corresponding word class sets. The sets obtained for the given text segments are illustrated in Figure 1.

Starting with each of the two text segments, and for each word in its word class sets, we determine the most similar word from the corresponding set in the other text segment. As mentioned earlier, we seek a WordNet-based semantic similarity for nouns and verbs, and only lexical matching for adjectives, adverbs, and cardinals. The word semantic similarity scores computed starting with the first text segment are shown in Table 3.

Text 1	Text 2	maxSim	IDF
jurors	jurors	1.00	5.80
courtroom	jurors	0.30	5.23
questionnaire	questionnaire	1.00	3.57
groups	questionnaire	0.29	0.85
were	were	1.00	0.09
taken	asked	1.00	0.28
asked	asked	1.00	0.45
fill	complete	0.86	1.29
out	–	0	0.06
40	–	0	1.39

Table 1: Wu & Palmer word similarity scores for computing text similarity with respect to text 1

Next, we use equation 7 and determine the semantic similarity of the two text segments with respect to text 1 as 0.6702, and with respect to text 2 as 0.7202. Finally, the two figures are combined into a bidirectional measure of similarity, calculated as 0.6952 based on equation 8.

Although there are a few words that occur in both text segments (e.g. *juror*, *questionnaire*), there are also words that are not identical, but closely related, e.g. *courtroom* found similar to *juror*, or *fill* which is related to *complete*. Unlike traditional similarity measures based on lexical matching, our metric takes into account the semantic similarity of these words, resulting in a more precise measure of text similarity.

4 Evaluation

To test the effectiveness of the text semantic similarity metric, we use this measure to automatically identify if two text segments are paraphrases of each other. We use the Microsoft paraphrase corpus (Dolan et al., 2004), consisting of 4,076 training pairs and 1,725 test pairs, and determine the number of correctly identified paraphrase pairs in the corpus using the text semantic similarity measure as the only indicator of paraphrasing. In addition, we also evaluate the measure using the PASCAL corpus (Dagan et al., 2005), consisting of 1,380 test–hypothesis pairs with a directional entailment (580 development pairs and 800 test pairs).

For each of the two data sets, we conduct two evaluations, under two different settings: (1) An unsupervised setting, where the decision on what constitutes a paraphrase (entailment) is made using a constant similarity threshold of 0.5 across all experiments; and (2) A supervised setting, where the optimal threshold and weights associated with various similarity metrics are determined through learning on training data. In this case, we use a voted perceptron algorithm (Freund and Schapire, 1998)[3].

We evaluate the text similarity metric built on top of the various word-to-word metrics introduced in Section 2.1. For comparison, we also compute three baselines: (1) A random baseline created by randomly choosing a true or false value for each text pair; (2) A lexical matching baseline, which only

[3] Classification using this algorithm was determined optimal empirically through experiments.

counts the number of matching words between the two text segments, while still applying the weighting and normalization factors from equation 7; and (3) A vectorial similarity baseline, using a cosine similarity measure as traditionally used in information retrieval, with *tf.idf* term weighting. For comparison, we also evaluated the corpus-based similarity obtained through LSA; however, the results obtained were below the lexical matching baseline and are not reported here.

For paraphrase identification, we use the bidirectional similarity measure, and determine the similarity with respect to each of the two text segments in turn, and then combine them into a bidirectional similarity metric. For entailment identification, since this is a directional relation, we only measure the semantic similarity with respect to the *hypothesis* (the text that is entailed).

We evaluate the results in terms of accuracy, representing the number of correctly identified true or false classifications in the test data set. We also measure precision, recall and F-measure, calculated with respect to the *true* values in each of the test data sets.

Tables 2 and 3 show the results obtained in the unsupervised setting, when a text semantic similarity larger than 0.5 was considered to be an indicator of paraphrasing (entailment). We also evaluate a metric that combines all the similarity measures using a simple average, with results indicated in the $Combined$ row.

The results obtained in the supervised setting are shown in Tables 4 and 5. The optimal combination of similarity metrics and optimal threshold are now determined in a learning process performed on the training set. Under this setting, we also compute an additional baseline, consisting of the most frequent label, as determined from the training data.

5 Discussion and Conclusions

For the task of paraphrase recognition, incorporating semantic information into the text similarity measure increases the likelihood of recognition significantly over the random baseline and over the lexical matching baseline. In the unsupervised setting, the best performance is achieved using a method that combines several similarity metrics into one, for an overall accuracy of 68.8%. When learning is used to find the optimal combination of metrics and optimal threshold, the highest accuracy of 71.5% is obtained by combining the similarity metrics and the lexical matching baseline together.

Metric	Acc.	Prec.	Rec.	F
Semantic similarity (knowledge-based)				
J & C	0.683	0.724	0.846	0.780
L & C	0.680	0.724	0.838	0.777
Lesk	0.680	0.724	0.838	0.777
Lin	0.679	0.717	0.855	0.780
W & P	0.674	0.722	0.831	0.773
Resnik	0.672	0.725	0.815	0.768
Combined	**0.688**	0.741	0.817	0.777
Baselines				
LexMatch	0.661	0.722	0.798	0.758
Vectorial	0.654	0.716	0.795	0.753
Random	0.513	0.683	0.500	0.578

Table 2: Text semantic similarity for paraphrase identification (unsupervised)

Metric	Acc.	Prec.	Rec.	F
Semantic similarity (knowledge-based)				
J & C	0.573	0.543	0.908	0.680
L & C	0.569	0.543	0.870	0.669
Lesk	0.568	0.542	0.875	0.669
Resnik	0.565	0.541	0.850	0.662
Lin	0.563	0.538	0.878	0.667
W & P	0.558	0.534	0.895	0.669
Combined	**0.583**	0.561	0.755	0.644
Baselines				
LexMatch	0.545	0.530	0.795	0.636
Vectorial	0.528	0.525	0.588	0.555
Random	0.486	0.486	0.493	0.489

Table 3: Text semantic similarity for entailment identification (unsupervised)

For the entailment data set, although we do not explicitly check for entailment, the directional similarity computed for textual entailment recognition does improve over the random and lexical matching baselines. Once again, the combination of similarity metrics gives the highest accuracy, measured at 58.3%, with a slight improvement observed in the supervised setting, where the highest accuracy was measured at 58.9%. Both these figures are competitive with the best results achieved during the PASCAL entailment evaluation (Dagan et al., 2005).

Although our method relies on a bag-of-words approach, as it turns out the use of measures of *semantic* similarity improves significantly over the traditional lexical matching metrics[4]. We are nonetheless

[4] The improvement of the combined semantic similarity metric over the simpler lexical matching measure was found to be statistically significant in all experiments, using a paired t-test ($p < 0.001$).

Metric	Acc.	Prec.	Rec.	F
Semantic similarity (knowledge-based)				
Lin	0.702	0.706	0.947	0.809
W & P	0.699	0.705	0.941	0.806
L & C	0.699	0.708	0.931	0.804
J & C	0.699	0.707	0.935	0.805
Lesk	0.695	0.702	0.929	0.800
Resnik	0.692	0.705	0.921	0.799
Combined	**0.715**	0.723	0.925	0.812
Baselines				
LexMatch	0.671	0.693	0.908	0.786
Vectorial	0.665	0.665	1.000	0.799
Most frequent	0.665	0.665	1.000	0.799

Table 4: Text semantic similarity for paraphrase identification (supervised)

Metric	Acc.	Prec.	Rec.	F
Semantic similarity (knowledge-based)				
L & C	0.583	0.573	0.650	0.609
W & P	0.580	0.570	0.648	0.607
Resnik	0.579	0.572	0.628	0.598
Lin	0.574	0.568	0.620	0.593
J & C	0.575	0.566	0.643	0.602
Lesk	0.573	0.566	0.633	0.597
Combined	**0.589**	0.579	0.650	0.612
Baselines				
LexMatch	0.568	0.573	0.530	0.551
Most frequent	0.500	0.500	1.000	0.667
Vectorial	0.479	0.484	0.645	0.553

Table 5: Text semantic similarity for entailment identification (supervised)

aware that a bag-of-words approach ignores many of important relationships in sentence structure, such as dependencies between words, or roles played by the various arguments in the sentence. Future work will consider the investigation of more sophisticated representations of sentence structure, such as first order predicate logic or semantic parse trees, which should allow for the implementation of more effective measures of text semantic similarity.

References

A. Budanitsky and G. Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Proceedings of the NAACL Workshop on WordNet and Other Lexical Resources*, Pittsburgh, June.

I. Dagan, O. Glickman, and B. Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Workshop*.

W.B. Dolan, C. Quirk, and C. Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland.

Y. Freund and R.E. Schapire. 1998. Large margin classification using the perceptron algorithm. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 209–217, New York, NY. ACM Press.

J. Jiang and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan.

T. K. Landauer, P. Foltz, and D. Laham. 1998. Introduction to latent semantic analysis. *Discourse Processes*, 25.

C. Leacock and M. Chodorow. 1998. Combining local context and WordNet sense similiarity for word sense disambiguation. In *WordNet, An Electronic Lexical Database*. The MIT Press.

M.E. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the SIGDOC Conference 1986*, Toronto, June.

C.Y. Lin and E.H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, PA, July.

S. Patwardhan, S. Banerjee, and T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February.

P. Resnik. 1995. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada.

J. Rocchio, 1971. *Relevance feedback in information retrieval*. Prentice Hall, Ing. Englewood Cliffs, New Jersey.

G. Salton and M.E. Lesk, 1971. *Computer evaluation of indexing and text processing*, pages 143–180. Prentice Hall, Ing. Englewood Cliffs, New Jersey.

G. Salton, , and A. Bukley. 1997a. Term weighting approaches in automatic text retrieval. In *Readings in Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA.

G. Salton, A. Singhal, M. Mitra, and C. Buckley. 1997b. Automatic text structuring and summarization. *Information Processing and Management*, 2(32).

K. Sparck-Jones. 1972. A statistical interpretation of term specificity and its applicatoin in retrieval. *Journal of Documentation*, 28(1):11–21.

E. Voorhees. 1993. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference*, Pittsburgh, PA.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico.

J. Xu and W. B. Croft. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference*, Zurich, Switzerland.