University_Of_Sheffield: Two Approaches to Semantic Text Similarity

Sam Biggins, Shaabi Mohammed, Sam Oakley,
Luke Stringer, **Mark Stevenson** and **Judita Priess**
Department of Computer Science
University of Sheffield
Sheffield
S1 4DP, UK
{aca08sb, aca08sm, coa07so, aca08ls,
r.m.stevenson, j.preiss}@shef.ac.uk

Abstract

This paper describes the University of Sheffield's submission to SemEval-2012 Task 6: Semantic Text Similarity. Two approaches were developed. The first is an unsupervised technique based on the widely used vector space model and information from WordNet. The second method relies on supervised machine learning and represents each sentence as a set of n-grams. This approach also makes use of information from WordNet. Results from the formal evaluation show that both approaches are useful for determining the similarity in meaning between pairs of sentences with the best performance being obtained by the supervised approach. Incorporating information from WordNet also improves performance for both approaches.

1 Introduction

This paper describes the University of Sheffield's submission to SemEval-2012 Task 6: Semantic Text Similarity (Agirre et al., 2012). The task is concerned with determining the degree of semantic equivalence between a pair of sentences.

Measuring the similarity between sentences is an important problem that is relevant to many areas of language processing, including the identification of text reuse (Seo and Croft, 2008; Bendersky and Croft, 2009), textual entailment (Szpektor et al., 2004; Zanzotto et al., 2009), paraphrase detection (Barzilay and Lee, 2003; Dolan et al., 2004), Information Extraction/Question Answering (Lin and Pantel, 2001; Stevenson and Greenwood, 2005), Information Retrieval (Baeza-Yates and Ribeiro-Neto, 1999), short answer grading (Pulman and Sukkarieh, 2005; Mohler and Mihalcea, 2009), recommendation (Tintarev and Masthoff, 2006) and evaluation (Papineni et al., 2002; Lin, 2004).

Many of the previous approaches to measuring the similarity between texts have relied purely on lexical matching techniques, for example (Baeza-Yates and Ribeiro-Neto, 1999; Papineni et al., 2002; Lin, 2004). In these approaches the similarity of texts is computed as a function of the number of matching tokens, or sequences of tokens, they contain. However, this approach fails to identify similarities when the same meaning is conveyed using synonymous terms or phrases (for example, "The dog sat on the mat" and "The hound sat on the mat") or when the meanings of the texts are similar but not identical (for example, "The cat sat on the mat" and "A dog sat on the chair").

Significant amounts of previous work on text similarity have focussed on comparing the meanings of texts longer than a single sentence, such as paragraphs or documents (Baeza-Yates and Ribeiro-Neto, 1999; Seo and Croft, 2008; Bendersky and Croft, 2009). The size of these texts means that there is a reasonable amount of lexical items in each document that can be used to determine similarity and failing to identify connections between related terms may not be problematic. The situation is different for the problem of semantic text similarity where the texts are short (single sentences). There are fewer lexical items to match in this case, making it more important that connections between related terms are identified. One way in which this information has been incorporated in NLP systems has

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 655–661,
Montréal, Canada, June 7-8, 2012.

been to make use of WordNet to provide information about similarity between word meanings, and this approach has been shown to be useful for computing text similarity (Mihalcea and Corley, 2006; Mohler and Mihalcea, 2009).

This paper describes two approaches to the semantic text similarity problem that use WordNet (Miller et al., 1990) to provide information about relations between word meanings. The two approaches are based on commonly used techniques for computing semantic similarity based on lexical matching. The first is unsupervised while the other requires annotated data to train a learning algorithm. Results of the SemEval evaluation show that the supervised approach produces the best overall results and that using the information provided by WordNet leads to an improvement in performance.

The remainder of this paper is organised as follows. The next section describes the two approaches for computing semantic similarity between pairs of sentences that were developed. The system submitted for the task is described in Section 3 and its performance in the official evaluation in Section 4. Section 5 contains the conclusions and suggestions for future work.

2 Computing Semantic Text Similarity

Two approaches for computing semantic similarity between sentences were developed. The first method, described in Section 2.1, is unsupervised. It uses an enhanced version of the vector space model by calculating the similarity between word senses, and then finding the distances between vectors constructed using these distances. The second method, described in Section 2.2, is based on supervised machine learning and compares sentences based on the overlap of the *n*-grams they contain.

2.1 Vector Space Model

The first approach is inspired by the vector space model (Salton et al., 1975) commonly used to compare texts in Information Retrieval and Natural Language Processing (Baeza-Yates and Ribeiro-Neto, 1999; Manning and Schütze, 1999; Jurafsky and Martin, 2009).

2.1.1 Creating vectors

Each sentence is tokenised, stop words removed and the remaining words lemmatised using NLTK (Bird et al., 2009). (The `WordPunctTokenizer` and `WordNetLemmatizer` are applied.) Binary vectors are then created for each sentence.

The similarity between sentences can be computed by comparing these vectors using the cosine metric. However, this does not take account of words with similar meanings, such as "dog" and "hound" in the sentences "The dog sat on the mat" and "The hound sat on the mat". To take account of these similarities WordNet-based similarity measures are used (Patwardhan and Pedersen, 2006).

Any terms that occur in only one of the sentences do not contribute to the similarity score since they will have a 0 value in the binary vector. Any words with a 0 value in one of the binary vectors are compared with all of the words in the other sentence and the similarity values computed. The highest similarity value is selected and use to replace the 0 value in that vector, see Figure 1. (If the similarity score is below the set threshold of 0.5 then the similarity value is not used and in these cases the 0 value remains unaltered.) This substitution of 0 values in the vectors ensures that similarity between words can be taken account of when computing sentence similarity.

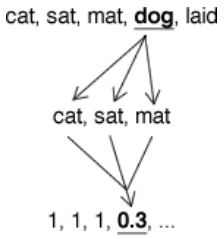

Figure 1: Determining word similarity values for vectors

Various techniques were explored for determining the similarity values between words. These are described and evaluated in Section 2.1.3.

2.1.2 Computing Sentence Similarity

The similarity between two sentences is computed using the cosine metric. Since the cosine metric is a distance measure, which returns a score of 0 for identical vectors, its complement is used to pro-

duce the similarity score. This score is multiplied by 5 in order to generate a score in the range required for the task.

2.1.3 Computing Word Similarity

The similarity values for the vectors are computed by first disambiguating each sentence and then applying a similarity measure. Various approaches for carrying out these tasks were explored.

Word Sense Disambiguation Two simple and commonly used techniques for Word Sense Disambiguation were applied.

- **Most Frequent Sense (MFS)** simply selects the first sense in WordNet, i.e., the most common occurring sense for the word. This approach is commonly used as a baseline for word sense disambiguation (McCarthy et al., 2004).
- **Lesk (1986)** chooses a synset by comparing its definition against the sentence and selecting the one with the highest number of words in common.

Similarity measures WordNet-based similarity measures have been found to perform well when used in combination with text similarity measures (Mihalcea and Corley, 2006) and several of these were compared. Implementations of these measures from the NLTK (Bird et al., 2009) were used.

- **Path Distance** uses the length of the shortest path between two senses to determine the similarity between them.
- **Leacock and Chodorow (1998)** expand upon the path distance similarity measure by scaling the path length by the maximum depth of the WordNet taxonomy.
- **Resnik (1995)** makes use of techniques from Information Theory. The measure of relatedness between two concepts is based on the Information Content of the Least Common Subsumer.
- **Jiang and Conrath (1997)** also uses the Information Content of the two input synsets.
- **Lin (1998)** uses the same values as Jiang and Conrath (1997) but takes the ratio of the shared information content to that of the individual concepts.

Results produced by the various combinations of word sense disambiguation strategy and similarity measures are shown in Table 1. This table shows the Pearson correlation of the system output with the gold standard over all of the SemEval training data. The row labelled 'Binary' shows the results using binary vectors which are not augmented with any similarity values. The remainder of the table shows the performance of each of the similarity measures when the senses are selected using the two word sense disambiguation algorithms.

Metric	MFS	Lesk
Binary	0.657	
Path Distance	0.675	0.669
Leacock and Chodorow (1998)	0.087	0.138
Resnik (1995)	0.158	0.153
Jiang and Conrath (1997)	0.435	0.474
Lin (1998)	0.521	0.631

Table 1: Performance of Vector Space Model using various disambiguation strategies and similarity measures

The results in this table show that the only similarity measure that leads to improvement above the baseline is the path measure. When this is applied there is a modest improvement over the baseline for each of the word sense disambiguation algorithms. However, all other similarity measures lead to a drop in performance. Overall there seems to be little difference between the performance of the two word sense disambiguation algorithms. The best performance is obtained using the paths distance and MFS disambiguation.

Table 2 shows the results of the highest scoring method broken down by the individual corpora used for the evaluation. There is a wide range between the highest (0.726) and lowest (0.485) correlation scores with the best performance being obtained for the MSRvid corpus which contains short, simple sentences.

Metric	Correlation
MSRpar	0.591
MSRvid	0.726
SMTeuroparl	0.485

Table 2: Correlation scores across individual corpora using Path Distance and Most Frequent Sense.

2.2 Supervised Machine Learning

For the second approach the sentences are represented as sets of *n*-grams of varying length, a common approach in text comparison applications which preserves some information about the structure of the document. However, like the standard vector space model (Section 2.1) this technique also fails to identify similarity between texts when an alternative choice of lexical item is used to express the same, or similar, meaning. To avoid this problem WordNet is used to generate sets of alternative *n*-grams. After the *n*-grams have been generated for each sentence they are augmented with semantic alternatives created using WordNet (Section 2.2.1). The overlap scores between the *n*-grams from the two sentences are used as features for a supervised learning algorithm (Section 2.2.2).

2.2.1 Generating *n*-grams

Preprocessing is carried out using NLTK. Each sentence is tokenised, lemmatised and stop words removed. A set of *n*-grams are then extracted from each sentence. The set of n-grams for the sentence S is referred to as S_o.

For every *n*-gram in S_o a list of alternative *n*-grams is generated using WordNet. Each item in the *n*-gram is considered in turn and checked to determine whether it occurs in WordNet. If it does then a set of alternative lexical items is constructed by combining all terms that are found in all synsets containing that item as well as their immediate hypernyms and hyponyms of the terms. An additional *n*-gram is created for each item in this set of alternative lexical items by substituting each for the original term. This set of expanded *n*-grams is referred to as S_a.

2.2.2 Sentence Comparison

Overlap metrics to determine the similarity between the sets of *n*-grams are used to create features for the learning algorithm. For two sentences, $S1$ and $S2$, four sets of *n*-grams are compared: $S1_o$, $S2_o$, $S1_a$ and $S2_a$ (i.e., the *n*-grams extracted directly from sentences $S1$ and $S2$ as well as the modified versions created using WordNet).

The *n*-grams that are generated using WordNet (S_a) are not as important as the original *n*-grams (S_o) for determining the similarity between sentences and this is accounted for by generating three different scores reflecting the overlap between the two sets of *n*-grams for each sentence. These scores can be expressed using the following equations:

$$\frac{|S1_o \cap S2_o|}{\sqrt{|S1_o| \times |S2_o|}} \quad (1)$$

$$\frac{|(S1_o \cap S2_a) \cap (S2_o \cap S1_a)|}{\sqrt{|(S1_o \cap S2_a)| \times |(S2_o \cap S1_a)|}} \quad (2)$$

$$\frac{|S1_a \cap S2_a|}{\sqrt{|S1_a| \times |S2_a|}} \quad (3)$$

Equation 1 is the cosine measure applied to the two sets of original n-grams, equation 2 compares the original *n*-grams in each sentence with the alternative *n*-grams in the other while equation 3 compares the alternative n-grams with each other.

Other features are used in addition to these similarity scores: the mean length of $S1$ and $S2$, the difference between the lengths of $S1$ and $S2$ and the corpus label (indicating which part of the SemEval training data the sentence pair was drawn from). We found that these additional features substantially increase the performance of our system, particularly the corpus label.

3 University of Sheffield's entry for Task 6

Our entry for this task consisted of three runs using the two approaches described in Section 2.

Run 1: Vector Space Model (VS) The first run used the unsupervised vector space approach (Section 2.1). Comparison of word sense disambiguation strategies and semantic similarity measures on the training data showed that the best results were obtained using the Path Distance Measure combined

with the Most Frequent Sense approach (see Tables 1 and 2) and these were used for the official run. Post evaluation analysis also showed that this strategy produced the best performance on the test data.

Run 2: Machine Learning (NG) The second run used the supervised machine learning approach (Section 2.2.2). The various parameters used by this approach were explored using 10-fold cross-validation applied to the SemEval training data. We varied the lengths of the *n*-grams generated, experimented with various pre-processing strategies and machine learning algorithms. The best performance was obtained using short *n*-grams, unigrams and bigrams, and these were used for the official run. Including longer *n*-grams did not lead to any improvement in performance but created significant computational cost due to the number of alternative *n*-grams that were created using WordNet. When the pre-processing strategies were compared it was found that the best performance was obtained by applying both stemming and stop word removal before creating *n*-grams and this approach was used in the official run. The Weka[1] `LinearRegression` algorithm was used for the official run and a single model was created by training on all of the data provided for the task.

Run 3: Hybrid (VS + NG) The third run is a hybrid combination of the two methods. The supervised approach (NG) was used for the three data sets that had been made available in the training data (MSRpar, MSRvid and SMT-eur) while the vector space model (VS) was used for the other two data sets. This strategy was based on analysis of performance of the two approaches on the training data. The NG approach was found to provide the best performance. However it was sensitive to the data set from which the training data was obtained from while VS, which does not require training data, is more robust.

A diagram depicting the various components of the submitted entry is shown in Figure 2.

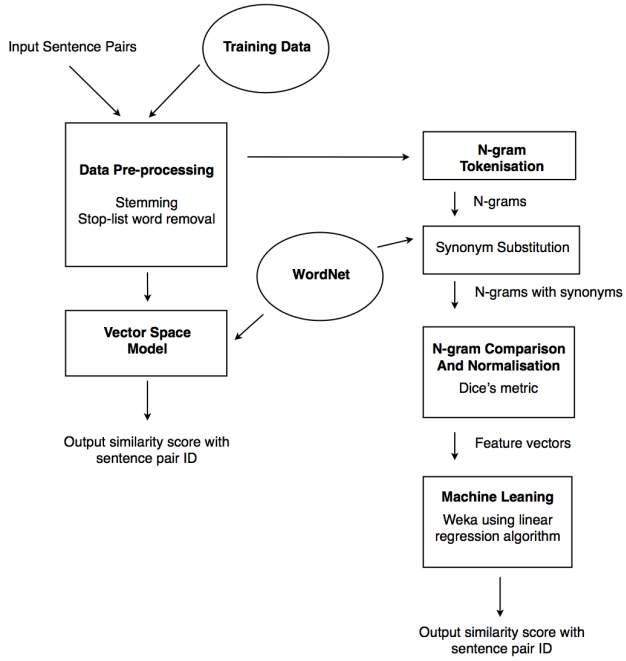

Figure 2: System Digram for entry

4 Evaluation

The overall performance (ALLnrm) of NG, VG and the hybrid systems is significantly higher than the official baseline (see Table 3). The table also includes separate results for each of the evaluation corpora (rows three to seven): the unsupervised VS model performance is significantly higher than the baseline (p-value of 0.06) over all corpus types, as is that of the hybrid model.

However, the performance of the supervised NG model is below the baseline for the (unseen in training data) SMT-news corpus. Given a pair of sentences from an unknown source, the algorithm employs a model trained on all data combined (i.e., omits the corpus information), which may resemble the input (On-WN) or it may not (SMT-news).

After stoplist removal, the average sentence length within MSRvid is 4.5, whereas it is 6.0 and 6.9 in MSRpar and SMT-eur respectively, and thus the last two corpora are expected to form better training data for each other. The overall performance on the MSRvid data is higher than for the other corpora, which may be due to the small number of adjectives and the simpler structure of the shorter sentences within the corpus.

The hybrid system, which selects the supervised system (NG)'s output when the test sentence pair is drawn from a corpus within the training data

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

Corpus	Baseline	Vector Space (VS)	Machine Learning (NG)	Hybrid (NG+VS)
ALL	.3110	.6054	.7241	.6485
ALLnrm	.6732	.7946	.8169	.8238
MSRpar	.4334	.5460	.5166	.5166
MSRvid	.2996	.7241	.8187	.8187
SMT-eur	.4542	.4858	.4859	.4859
On-WN	.5864	.6676	.6390	.6676
SMT-news	.3908	.4280	.2089	.4280

Table 3: Correlation scores from official SemEval results

Rank (/89)	Rank	Ranknrm	RankMean
Baseline	87	85	70
Vector Space (VS)	48	44	29
Machine Learning (NG)	17	18	37
Hybrid	34	15	20

Table 4: Ranks from official SemEval results

and selects the unsupervised system (VS)'s answer otherwise, outperforms both systems in combination. Contrary to expectations, the supervised system did not always outperform VS on phrases based on training data – the performance of VS on MSRpar, with its long and complex sentences, proved to be slightly higher than that of NG. However, the unsupervised system was clearly the correct choice when the source was unknown.

5 Conclusion and Future Work

Two approaches for computing semantic similarity between sentences were explored. The first, unsupervised approach, uses a vector space model and computes similarity between sentences by comparing vectors while the second is supervised and represents the sentences as sets of n-grams. Both approaches used WordNet to provide information about similarity between lexical items. Results from evaluation show that the supervised approach provides the best results on average but also that performance of the unsupervised approach is better for some data sets. The best overall results for the SemEval evaluation were obtained using a hybrid system that attempts to choose the most suitable approach for each data set.

The results reported here show that the semantic text similarity task can be successfully approached using lexical overlap techniques augmented with limited semantic information derived from WordNet. In future, we would like to explore whether performance can be improved by applying deeper analysis to provide information about the structure and semantics of the sentences being compared. For example, parsing the input sentences would provide more information about their structure than can be obtained by representing them as a bag of words or set of n-grams. We would also like to explore methods for improving performance of the n-gram overlap approach and making it more robust to different data sets.

Acknowledgements

This research has been supported by a Google Research Award.

References

E. Agirre, D. Cer, M Diab, and A. Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), in conjunction with the First Joint Conference on Lexical and Computational Semantics (*SEM 2012)*.

R. Baeza-Yates and B. Ribeiro-Neto. 1999. *Modern Information Retrieval*. Addison Wesley Longman Limited, Essex.

R. Barzilay and L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

M. Bendersky and W.B. Croft. 2009. Finding text reuse on the web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 262–271. ACM.

S. Bird, E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

B. Dolan, C. Quirk, and C. Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of Coling 2004*, pages 350–356, Geneva, Switzerland.

J.J. Jiang and D.W. Conrath. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*.

D. Jurafsky and J. Martin. 2009. *Speech and Language Processing*. Pearson, second edition.

C. Leacock and M. Chodorow, 1998. *Combining local context and WordNet similarity for word sense identification*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

M. Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of ACM SIGDOC Conference*, pages 24–26, Toronto, Canada.

D. Lin and P. Pantel. 2001. Discovery of interence rules for question answering. *Natural Language Engineering*, 7(4):343–360.

D. Lin. 1998. An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304.

C. Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July.

C. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL-2004)*, pages 280–287, Barcelona, Spain.

R. Mihalcea and C. Corley. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *In AAAI06*, pages 775–780.

G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. 1990. WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–312.

M. Mohler and R. Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 567–575, Athens, Greece.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

S. Patwardhan and T. Pedersen. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concept. In *Proceedings of the workshop on "Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together" held in conjunction with the EACL 2006*, pages 1–8.

S.G. Pulman and J.Z. Sukkarieh. 2005. Automatic short answer marking. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, pages 9–16, Ann Arbor, Michigan.

P. Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453.

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620.

J. Seo and W.B. Croft. 2008. Local text reuse detection. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 571–578.

M. Stevenson and M. Greenwood. 2005. A Semantic Approach to IE Pattern Induction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 379–386, Ann Arbour, MI.

I. Szpektor, H. Tanev, I. Dagan, and B. Coppola. 2004. Scaling web-based acquisition of entailment relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 41–48, Barcelona, Spain.

N. Tintarev and J. Masthoff. 2006. Similarity for news recommender systems. In *In Proceedings of the AH'06 Workshop on Recommender Systems and Intelligent User Interfaces*.

F.M. Zanzotto, M. Pennacchiotti, and A. Moschitti. 2009. A machine learning approach to textual entailment recognition. *Natural Language Engineering*, 15-04:551–582.