sranjans : Semantic Textual Similarity using Maximal Weighted Bipartite Graph Matching

Sumit Bhagwani, Shrutiranjan Satapathy, Harish Karnick
Computer Science and Engineering
IIT Kanpur, Kanpur - 208016, India
{sumitb,sranjans,hk}@cse.iitk.ac.in

Abstract

The paper aims to come up with a system that examines the degree of semantic equivalence between two sentences. At the core of the paper is the attempt to grade the similarity of two sentences by finding the maximal weighted bipartite match between the tokens of the two sentences. The tokens include single words, or multiwords in case of Named Entitites, adjectivally and numerically modified words. Two token similarity measures are used for the task - WordNet based similarity, and a statistical word similarity measure which overcomes the shortcomings of WordNet based similarity. As part of three systems created for the task, we explore a simple *bag of words* tokenization scheme, a more careful tokenization scheme which captures named entities, times, dates, monetary entities etc., and finally try to capture context around tokens using grammatical dependencies.

1 Introduction

Semantic Textual Similarity (STS) measures the degree of semantic equivalence between texts. The goal of this task is to create a unified framework for the evaluation of semantic textual similarity modules and to characterize their impact on NLP applications. The task is part of the Semantic Evaluation 2012 Workshop (Agirre et al., 2012).

STS is related to both Textual Entailment and Paraphrase, but differs in a number of ways and it is more directly applicable to a number of NLP tasks. Also, STS is a graded similarity notion - this graded bidirectional nature of STS is useful for NLP tasks such as MT evaluation, information extraction, question answering, and summarization.

We propose a lexical similarity approach to grade the similarity of two sentences, where a maximal weighted bipartite match is found between the tokens of the two sentences. The approach is robust enough to apply across different datasets. The results on the STS test datasets are encouraging to say the least. The tokens are single word tokens in case of the first system, while in the second system, named and monetary entities, percentages, dates and times are handled too. A token-token similarity measure is integral to the approach and we use both a statistical similarity measure and a WordNet based word similarity measure for the same. In the final run of the task, apart from capturing the aforementioned entities, we heuristically extract adjectivally and numerically modified words. Also, the last run naively attempts to capture the context around the tokens using grammatical dependencies, which in turn is used to measure context similarity.

Section 2 discusses the previous work done in this area. Section 3 describes the datasets, the baseline system and the evaluation measures used by the task organizers. Section 4, 5 and 6 introduce the systems developed and discuss the results of each system. Finally, section 7 con-

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 579–585,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

cludes the work and section 8 offers suggestions for future work.

2 Related Work

Various systems exist in literature for textual similarity measurement, be it bag of words based models or complex semantic systems. (Achananuparp et al., 2008) enumerates a few word overlap measures, like Jaccard Similarity Coefficient, IDF Overlap measures, Phrasal overlap measures etc, that have been used for sentential similarity.

(Liu et al., 2008) proposed an approach to calculate sentence similarity, which takes into account both semantic information and word order. They define semantic similarity of *sentence 1* relative to *sentence 2* as the ratio of the sum of the word similarity weighted by information content of words in *sentence 1* to the overall information content included in both sentences. The syntactic similarity is calculated as the correlation coefficient between word order vectors.

A similar semantic similarity measure, proposed by (Li et al., 2006), uses a semantic-vector approach to measure sentence similarity. Sentences are transformed into feature vectors having individual words from the sentence pair as a feature set. Term weights are derived from the maximum semantic similarity score between words in the feature vector and words in the corresponding sentence. To utilize word order in the similarity calculation, they define a word order similarity measure as the normalized difference of word order between the two sentences. They have empirically proved that a sentence similarity measure performs the best when semantic measure is weighted more than syntactic measure (ratio $\sim$ 4:1). This follows the conclusion from a psychological experiment conducted by them which emphasizes the role of semantic information over syntactic information in passage understanding.

3 Task Evaluation

3.1 Datasets

The development datasets are drawn from the following sources :

- **MSR Paraphrase :** This dataset consists of pairs of sentences which have been extracted from news sources on the web.
- **MSR Video :** This dataset consists of pairs of sentences where each sentence of a pair tries to summarize the action in a short video snippet.
- **SMT Europarl :** This dataset consists of pairs sentences drawn from the proceedings of the European Parliament, where each sentence of a pair is a translation from a European language to English.

In addition to the above sources, the test datasets also contained the following sources :

- **SMT News :** This dataset consists of machine translated news conversation sentence pairs.
- **On WN :** This dataset consists of pairs of sentences where the first comes from Ontonotes(Hovy et al., 2006) and the second from a WordNet definition. Hence, the sentences are rather phrases.

3.2 Baseline

The task organizers have used the following baseline scoring scheme. Scores are produced using a simple word overlap baseline system. The input sentences are tokenised by splitting at white spaces, and then each sentence is represented as a vector in the multidimensional token space. Each dimension has 1 if the token is present in the sentence, 0 otherwise. Similarity of vectors is computed using the cosine similarity.

3.3 Evaluation Criteria

The scores obtained by the participating systems are evaluated against the gold standard of the datasets using a pearson correlation measure. In order to evaluate the overall performance of the systems on all the five datasets, the organizers use three evaluation measures :

- **ALL :** This measure takes the union of all the test datasets, and finds the Pearson correlation of the system scores with the gold standard of the union.

- **ALL Normalized :** In this measure, a linear fit is found for the system scores on each dataset using a least squared error criterion, and then the union of the linearly fitted scores is used to calculate the Pearson correlation against the gold standard union.
- **Weighted Mean :** The average of the Pearson correlation scores of the systems on the individual datasets is taken, weighted by the number of test instances in each dataset.

4 SYSTEM 1

4.1 Tokenization Scheme

Each sentence is tokenized into words, filtering out punctuations and stop-words. The stop-words are taken from the stop-word list provided by the NLTK Toolkit (Bird et al., 2009). All the word tokens are reduced to their lemmatized form using the Stanford CoreNLP Toolkit (Minnen et al., 2001). The tokenization is basic in nature and doesn't handle named entities, times, dates, monetary entities or multi-word expressions. The challenge with handling multi-word tokens is in calculating multi-word token similarity, which is not supported in a WordNet word-similarity scheme or a statistical word similarity measure.

4.2 Maximal Weighted Bipartite Match

A weighted bipartite graph is constructed where the two sets of vertices are the word-tokens extracted in the earlier subsection. The bipartite graph is made complete by assigning an edge weight to every pair of tokens from the two sentences. The edge weight is based on a suitable word similarity measure. We had two resources at hand - WordNet based word similarity and a statistical word similarity measure.

4.2.1 WordNet Based Word Similarity

The is-a hierarchy of WordNet is used in calculating the word similarity of two words. Nouns and verbs have separate is-a hierarchies. We use the Lin word-sense similarity measure (Lin , 1998a). Adjectives and adverbs do not have an is-a hierarchy and hence do not figure in the Lin similarity measure. To disambiguate the WordNet sense of a word in a sentence, a variant of the Simplified Lesk Algorithm (Kilgarriff and J. Rosenzweig , 2000) is used. WordNet based word similarity has the following drawbacks :

- sparse in named entity content : similarity of named entities with other words becomes infeasible to calculate.
- doesn't support cross-POS similarity.
- applicable only to nouns and verbs.

4.2.2 Statistical Word Similarity

We use DISCO (Kolb , 2008) as our statistical word similarity measure. DISCO is a tool for retrieving the distributional similarity between two given words. Pre-computed word spaces are freely available for a number of languages. We use the English Wikipedia word space. One primary reason for using a statistical word similarity measure is because of the shortcomings of calculating cross-POS word similarity when using a knowledge base like WordNet.

DISCO works as follows : a term-(term, relative position) matrix is constructed with weights being pointwise mutual information scores. From this, a surface level word similarity score is obtained by using Lin's information theoretic measure (Lin , 1998b) for word vector similarity. This score is used as matrix weights to get second order word vectors, which are used to compute a second order word similarity measure . This measure tries to emulate an LSA like similarity giving better performance, and hence is used for the task.

A point to note here is that the precomputed word spaces that DISCO uses are case sensitive, which we think is a drawback. We preserve the case of proper nouns, while all other words are converted to lower case, prior to evaluating word similarity scores.

4.3 Edge Weighting Scheme

Sentences in the MSR video dataset are simpler and shorter than the remaining datasets, with a high degree of POS correspondence between the

Dataset	DISCO	WordNet	DISCO + WordNet
MSR Video	0.61	0.71	0.73
MSR Paraphrase	0.62	0.43	0.57
SMT Europarl	0.58	0.44	0.54

Figure 1: Edge Weight Scheme Evaluation on Development Datasets

Category	NE	Normalized NE
DATE	26th November, November 26	XXXX-11-26
PERCENT	13 percent, 13%	%13.0
MONEY	56 dollars, $56, 56$	$56.0
TIME	3 pm, 15:00	T15:00

Figure 2: Normalization performed by Stanford CoreNLP

tokens of two sentences, as can be observed in the following example :

- *A man is riding a bicycle.* VS *A man is riding a bike.*

This allows for the use of a Knowledge-Base Word Similarity measure like WordNet word similarity. All the other datasets have lengthier sentences, resulting in cross-POS correspondence. Additionally, there is an abundance of named entities in these datasets. The following examples, which are drawn from the MSR Paraphrase dataset, highlight these points :

- *If convicted of the spying charges, he could face the death penalty.* VS *The charges of espionage and aiding the enemy can carry the death penalty.*

- *Microsoft has identified the freely distributed Linux software as one of the biggest threats to its sales.* VS *The company has publicly identified Linux as one of its biggest competitive threats.*

Keeping this in mind, we use DISCO for edge-weighting in all the datasets except MSR Video. For MSR Video, we use the following edge weighting scheme : for same-POS words, WordNet similarity is used, DISCO otherwise. This choice is justified by the results obtained in figure 1 on the development datasets.

4.3.1 Scoring

A maximal weighted bipartite match is found for the bipartite graph constructed, using the Hungarian Algorithm (Kuhn , 1955) - the intuition behind this being that every keyword in a sentence matches injectively to a unique keyword in the other sentence. The maximal bipartite score is normalized by the sentences' length for two reasons - normalization and punishment for extra detailing in either sentence. So the final sentence similarity score between sentences s_1 and s_2 is:

$$sim(s_1, s_2) = \frac{MaximalBipartiteMatchSum(s_1,s_2)}{max(tokens(s_1),tokens(s_2))}$$

4.4 Results

The results are evaluated on the test datasets provided for the STS task. Figure 3 compares the performance of our systems with the top 3 systems for the task. The scores in the figure are Pearson Correlation scores. Figure 4 shows the performance and ranks of all our systems. A total of 89 systems were submitted, including the baseline. The results are taken from the Semeval'12 Task 6 webpage[1]

As can be seen, System 1 suffers slightly on the MSR Paraphrase and Video datasets, while doing comparably well on the other three datasets when compared with the top 3 submissions. Our ALL score suffers because we use

[1] `http://www.cs.york.ac.uk/semeval-2012/task6/index.php?id=results-update`

System	ALL	MSR Paraphrase	MSR Video	SMT Europarl	OnWN	SMT News
Rank 1	0.8239	0.6830	0.8739	0.5280	0.6641	0.4937
Rank 2	0.8138	0.6985	0.8620	0.3612	0.7049	0.4683
Rank 3	0.8133	0.7343	0.8803	0.4771	0.6797	0.3989
System 1	0.6529	0.6124	0.7240	0.5581	0.6703	0.4533
System 2	0.6651	0.6254	0.7538	0.5328	0.6649	0.5036
System 3	0.5045	0.6167	0.7061	0.5666	0.5664	0.3968
Li et al.	0.4981	0.6141	0.6084	0.5382	0.6055	0.3760
Baseline	0.3110	0.4334	0.2996	0.4542	0.5864	0.3908

Figure 3: Results of top 3 Systems and Our Systems

System	ALL	ALL Rank	All Normalized	All Normalized Rank	Weighted Mean	Weighted Mean Rank
System 1	0.6529	30	0.8018	39	0.6249	12
System 2	0.6651	24	0.8128	22	0.6366	8
System 3	0.5045	62	0.7846	52	0.5905	30

Figure 4: Evaluation of our Systems on different criteria

a combination of WordNet and statistical word similarity measure for the MSR Video dataset, which affects the Pearson Correlation of all the datasets combined. The correlation values for the ALL Normalized criterion are high because of the linear fitting it performs. We get the best performance on the Weighted Mean evaluation criterion.

5 SYSTEM 2

In System 2, in addition to System 1, we capture named entities, dates and times, percentages and monetary entities and normalize them. The tokens resulting from this can be multi-word because of named entities. This tokenization strategy gives us the best results among all our three runs. For capturing and normalizing the above mentioned expressions, we make use of the Stanford NER Toolkit (Finkel et al., 2005). Some normalized samples are mentioned in figure 2.

When grading the similarity of multi-word tokens, we use a second level maximal bipartite match, which is normalized by the smaller of the two multi-word token lengths. Thus, similarity between two multi-word tokens t_1 and t_2 is defined as:
$sim(t_1, t_2) = \frac{MaximalBipartiteMatchSum(t_1,t_2)}{min(words(t_1),words(t_2))}$

This was done to ensure that a complete named entity in the first sentence matches exactly with a partial named entity (indicating the same entity as the first) in the second sentence. For eg. *John Doe vs John* will be given a score of 1. Such occurrences are frequent in the task datasets. For the sentence similarity, the score defined in System 1 is used, where the token length of a sentence is the number of multi-word tokens in it.

5.1 Results

Refer to figures 3 and 4 for results.

This system gives the best results among all our systems. The credit for this improvement can be attributed to recognition and normalization of named entities, dates and times, percentages and monetary entities, as the datasets provided contain these in fairly large numbers.

6 SYSTEM 3

In System 3, in addition to System 2, we heuristically capture compound nouns, adjectivally and numerically modified words like ’passenger plane’, ’easy job’, ’10 years’ etc. using the POS based regular expression

$$[JJ|NN|CD]^*NN$$

POS Tagging is done using the Stanford POS Tagger Toolkit (Toutanova et al., 2003).

To make matching more context dependent, rather than just a bag of words approach, we naively attempt to capture the similarity of the contexts of two tokens. We define the context of a word in a sentence as all the words in the sentence which are grammatically related to it. The grammatical relations are all the collapsed dependencies produced by the Stanford Dependency parser (Marneffe et al., 2006). The context of a multi-word token is defined as the union of contexts of all the words in it. We further filter the context by removing stop-words and punctuations in it. The contexts of two tokens are then used to obtain context/syntactic similarity between tokens, which is defined using the Jaccard Similarity Measure:

$$Jaccard(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

A linear combination of word similarity and context similarity is taken as an edge weight in the token-token similarity bipartite graph. Motivated by (Li et al., 2006), we chose a ratio of 4:1 for lexical similarity to context similarity.

As in System 2, for multi-word token similarity, we use a second level maximal bipartite match, normalized by smaller of the two token lengths. This helps in matching multi-word tokens expressing the same meaning with score 1, for e.g. *passenger plane* VS *Cuban plane*, *divided Supreme Court* VS *Supreme Court* etc. The sentence similarity score is the same as the one defined in System 2.

6.1 Results

Refer to figures 3 and 4 for results.

This system gives a reduced performance compared to our other systems. This could be due to various factors. Capturing adjectivally and numerically modified words could be done using grammatical dependencies instead of a heuristic POS-tag regular expression. Also, token-token similarity should be handled in a more precise way than a generic second level maximal bipartite match. A better context capturing method can further improve the system.

7 Conclusions

Among the three systems proposed for the task, System 2 performs best on the test datasets, primarily because it identifies named entities as single entities, normalizes dates, times, percentages and monetary figures. The results for System 3 suffer because of naive context capturing. A better job can be done using syntacto-semantic structured representations for the sentences. The performance of our systems are compared with (Li et al., 2006) on the test datasets in figure 3. This highlights the improvement of maximal weighted bipartite matching over greedy matching.

8 Future Work

Our objective is to group words together which share a common meaning. This includes grouping adjectival, adverbial, numeric modifiers with the modified word, group the words of a colloquial phrase together, capture multi-word expressions, etc. These word-clusters will form the vertices of the bipartite graph. The other challenge then is to come up with a suitable cluster-cluster similarity measure. NLP modules such as Lexical Substitution can help when we are using a word-word similarity measure at the core.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

Dan Klein and Christopher D. Manning. 2003. *Accurate Unlexicalized Parsing*. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw and Ralph Weischedel. 2006. *OntoNotes: The 90% Solution*. Proceedings of HLT/NAACL, New York, 2006.

Eneko Agirre, Daniel Cer, Mona Diab and Aitor 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).

G. Minnen, J. Carroll and D. Pearce. 2001. *Applied morphological processing of English*. Natural Language Engineering, 7(3). 207-223.

Harold W. Kuhn. 1955. *The Hungarian Method for the assignment problem*. Naval Research Logistics Quarterly, 2:8397, 1955.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370

Kilgarriff and J. Rosenzweig. 2000. *English SENSEVAL : Report and Results*. In Proceedings of the 2nd International Conference on Language Resources and Evaluation, LREC, Athens, Greece.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In Proceedings of HLT-NAACL 2003, pp. 252-259.

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

Lin, D. 1998b. *Automatic Retrieval and Clustering of Similar Words.*. In Proceedings of COLING-ACL 1998, Montreal.

Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. *Generating Typed Dependency Parses from Phrase Structure Parses*. In LREC 2006.

Palakorn Achananuparp, Xiaohua Hu and Shen Xiajiong. 2008. *The Evaluation of Sentence Similarity Measures*. Science And Technology, 5182, 305-316. Springer.

Peter Kolb. 2008. *DISCO: A Multilingual Database of Distributionally Similar Words*. In Proceedings of KONVENS-2008, Berlin.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009

Xiao-Ying Liu, Yi-Ming Zhou, Ruo-Shi Zheng. 2008. *Measuring Semantic Similarity Within Sentences*. Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming.

Yuhua Li, David McLean, Zuhair A. Bandar, James D. OShea, and Keeley Crockett. 2006. *Sentence Similarity Based on Semantic Nets and Corpus Statistics*. IEEE Transections on Knowledge and Data Engineering, Vol. 18, No. 8