SRIUBC: Simple Similarity Features for Semantic Textual Similarity

Eric Yeh
SRI International
Menlo Park, CA USA
yeh@ai.sri.com

Eneko Agirre
University of the Basque Country
Donostia, Basque Country
e.agirre@ehu.es

Abstract

We describe the systems submitted by SRI International and the University of the Basque Country for the Semantic Textual Similarity (STS) SemEval-2012 task. Our systems focused on using a simple set of features, featuring a mix of semantic similarity resources, lexical match heuristics, and part of speech (POS) information. We also incorporate precision focused scores over lexical and POS information derived from the BLEU measure, and lexical and POS features computed over split-bigrams from the ROUGE-S measure. These were used to train support vector regressors over the pairs in the training data. From the three systems we submitted, two performed well in the overall ranking, with split-bigrams improving performance over pairs drawn from the MSR Research Video Description Corpus. Our third system maintained three separate regressors, each trained specifically for the STS dataset they were drawn from. It used a multinomial classifier to predict which dataset regressor would be most appropriate to score a given pair, and used it to score that pair. This system underperformed, primarily due to errors in the dataset predictor.

1 Introduction

Previous semantic similarity tasks, such as paraphrase identification or recognizing textual entailment, have focused on performing binary decisions. These problems are usually framed in terms of identifying whether a pair of texts exhibit the needed similarity or entailment relationship or not. In many cases, such as producing a ranking over similarity scores, a soft measure of similarity between a pair of texts would be more desirable.

We contributed three systems for the 2012 Semantic Textual Similarity (STS) task (Agirre et al., 2012). These are:

1. **System 1**, which used a combination of semantic similarity, lexical similarity, and precision focused part-of-speech (POS) features.

2. **System 2**, which used features from System 1, with the addition of skip-bigram features derived from the ROUGE-S (Lin, 2004) measure. POS variants of skip-bigrams were incorporated as well.

3. **System 3**, used the features from above to first classify the dataset the pair was drawn from, and then applied regressors trained for that dataset.

Our systems characterize sentence pairs as feature vectors, populated by a variety of scorers that will be described below. During training, we used support vector regression (SVR) to train regressors against these vectors and their associated similarity scores.

The STS training data is divided into three datasets, reflecting their origin: Microsoft Research Paraphrase Corpus (MSRpar), MSR Research Video Description Corpus (MSRvid), and WMT2008 Development dataset (SMTeuroparl). We trained individual regressors for each of these datasets, and applied them to their counterparts in the testing set.

Both Systems 1 and 2 used the following types of features:

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 617–623,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

1. Resource based word to word semantic similarities.
2. Cosine-based lexical similarity measure.
3. Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) lexical overlap.
4. Precision focused Part of Speech (POS) features.

System 2 added the following features:

1. Lexically motivated skip-bigram overlap.
2. Precision focused skip-bigram POS features.

One of the primary motivations for our the choice of features was to use relatively simple and fast features, which can be scaled up to large datasets, given appropriate caching and pre-generated lookups. As the test phase included surprise datasets, whose origin was not disclosed, we also trained a fourth model using all of the training data from all three datasets. Systems 1 and 2 employed this strategy for the surprise data.

Since the statistics for each of the training datasets varied, directly pooling them together may not be the best strategy when scoring the surprise data, whose origins were unknown. To account for this, System 3 treated this as a gated regression problem, where pairs are considered to originate strictly from one dataset, and to score using a model specifically tailored for that dataset. We first trained regressors on each of the datasets separately. Then we trained a classifier to predict which dataset a given pair is likeliest to have been drawn from, and then applied the matching trained regressor to obtain its score.

This team included one of the organizers. We want to stress that we took all measures to make our participation on the same conditions as the rest of participants. In particular, the organizer did not allow the other member of the team to access any data or information which was not already available for the rest of participants.

For the rest of this system description, we first outline the scorers used to populate the feature vectors used for Systems 1 and 2. We then describe the setup for performing the regression. We follow with an explanation of our strategies for dealing with the surprise data, including a description of System 3. We then summarize performance over the the datasets, and discuss future avenues of investigation.

2 Resource Based Similarity

Our system uses several resources for assessing the word to word similarity between a pair of sentences. In order to pool together the similarity scores for a given pair, we employed the Semantic Matrix (Fernando and Stevenson, 2008) framework. To generate the scores, we used several resources, principally those derived from the relation graph of WordNet (Fellbaum, 1998), and those derived from distributional resources, namely Explicit Semantic Analysis (Gabrilovich and Markovitch, 2009), and the Dekang Lin Proximity-based Thesaurus [1]. We now describe the Semantic Matrix method, and follow with descriptions of each of the resources used.

2.1 Semantic Matrix

The Semantic Matrix is a method for pooling all of the pairwise similarity scores between the tokens found in two input strings. In order to score the similarity between a pair of strings s_1 and s_2 we first identify all of the unique vocabulary words from these strings to derive their corresponding occurrence vectors $\mathbf{v_1}$ and $\mathbf{v_2}$. Each dimension of these vectors corresponds to a unique vocabulary word, and binary values were used, corresponding to whether that word was observed. The similarity score for pair, $\mathrm{sim}(s_1, s_2)$, is given by Formula 1.

$$\mathrm{sim}(s_1, s_2) = \frac{\mathbf{v}_1^T \mathbf{W} \mathbf{v}_2}{\|\mathbf{v_1}\| \, \|\mathbf{v_2}\|} \tag{1}$$

with $\mathbf{W}$ being the symmetric matrix marking the similarity between pairs of words in the vocabulary. We note that this is similar to the Mahalanobis distance, except adjusted to produce a similarity. For this experiment, we normalized matrix entries so all values lay in the 0-1 range.

As named entities and other words encountered may not appear in one or more of the resources used, we applied the identity to $\mathbf{W}$. This is equivalent to adding a strict lexical match fallback on top of the similarity measure.

[1] http://webdocs.cs.ualberta.ca/ lindek/downloads.htm

Per (Fernando and Stevenson, 2008), a filter was applied over the values of $\mathbf{W}$. Any entries that fell below a given threshold value were flattened to zero, in order to prevent low scoring similarities from overwhelming the score. From previous studies over MSRpar, we applied a threshold of 0.9.

For our experiments, each of the word to word similarity scorers described below were used to generate a corresponding word similarity matrix $\mathbf{W}$, with scores generated using the Semantic Matrix.

2.2 WordNet Similarity

We used several methods to obtain word to word similarities from WordNet. WordNet is a lexical-semantic resource that describes typed relationships between *synsets*, semantic categories a word may belong to. Similarity scoring methods identify the synsets associated with a pair of words, and then use this relationship graph to generate a score.

The first set of scorers were generated from the Leacock-Chodorow, Lin, and Wu-Palmer measures from the WordNet Similarity package (Pedersen et al., 2004). For each of these measures, we averaged across all of the possible synsets between a given pair of words.

Another scorer we used was Personalized PageRank (PPR) (Agirre et al., 2010), a topic sensitive variant of the PageRank algorithm (Page et al., 1999) that uses a random walk process to identify the significant nodes of a graph given its link structure. We first derived a graph $\mathbf{G}$ from WordNet, treating synsets as the vertices and the relationships between synsets as the edges. To obtain a signature for a given word, we apply topic sensitive PageRank (Haveliwala, 2002) over $\mathbf{G}$, using the synsets associated with the word as the initial distribution. At convergence, we convert the stationary distribution into a vector. The similarity between two words is the cosine similarity between their vectors.

2.3 Distributional Resources

In contrast with the structure based WordNet based methods, distributional methods use statistical properties of corpora to derive similarity scores. We generated two scorers, one based on Explicit Semantic Analysis (ESA), and the other on the Dekang Lin Proximity-based Thesaurus. For a given word, ESA generates a *concept vector*, where the concepts are Wikipedia articles, and the score measures how closely associated that word is with the textual content of the article. To score the similarity between two words, we computed the cosine similarity of their concept vectors. This method proved to give state-of-the-art performance on the WordSim-353 word pair relatedness dataset (Finkelstein et al., 2002).

The Lin Proximity-based Thesaurus identifies the neighborhood around words encountered in the Reuters and Text Retrieval Conference (TREC). For a given word, the Thesaurus identifies the top 200 words with the most similar neighborhoods, listing the score based on these matches. For our experiments, we treated these as feature vectors, with the intuition being similar words should share similar neighbors. Again, the similarity score between two words was scored using the cosine similarity of their vectors.

3 Cosine Similarity

Another scorer we used was the cosine similarity over the lemmas found in the sentences in a pair. For generating the vectors used in the cosine similarity computation, we used the term frequency of the lemmas.

4 BLEU Features

BLEU is a measure developed to automatically assess how closely sentences generated by machine translation systems match reference human generated texts. BLEU is a directional measurement, and works on the assumption that the more lexically similar a *system* generated sentence is to a *reference* sentence, a human generated translation, the better the system sentence is. This can also be seen as a stand-in for the semantic similarity of the pairs, as was shown when BLEU was applied to the paraphrase identification identification problem in (Finch et al., 2005).

The BLEU score for a given system sentence and reference sentence of order N is computed using Formula 2.

$$\mathrm{BLEU}(sys, ref) = B \cdot \exp \sum_{n=1}^{N} \frac{1}{N} \log(p_n) \quad (2)$$

B is a brevity penalty used to prevent degenerate translations. Given this has little bearing on our experiments, we set its value to 1 for our experiments. Following (Papineni et al., 2002), we give each order n equal weight in the geometric mean. The probability of an order n-gram from the system sentence being found in the reference, p_n, is given in Formula 3.

$$p_n = \frac{\sum_{ngram \in sys} \text{count}_{sys \wedge ref}(ngram)}{\sum_{ngram \in sys} \text{count}_{sys}(ngram)} \tag{3}$$

$\text{count}_{sys}(ngram)$ is frequency of occurrence for the given n-gram in the system sentence. The numerator term is computed as $\text{count}_{sys \wedge ref}(ngram) = \min(\text{count}_{sys}(ngram), \text{count}_{ref}(ngram))$ where $\text{count}_{ref}(ngram)$ is the frequency of occurrence of that n-gram in the reference sentence. This is equivalent to having each n-gram have a 1-1 mapping with a matching n-gram in the reference (if any), and counting the number of mappings.

As there is a risk of higher order system n-grams having no matches in the reference, we apply Laplacian smoothing to the n-gram counts.

BLEU is considered to be a precision focused measure, as it only measures how much of the system sentence matches a reference sentence. Following (Finch et al., 2005), we obtain a modified BLEU score for strings s_1 and s_2 of a pair by averaging the BLEU scores where each takes a turn as the system sentence, as given in Formula 4.

$$\text{Score}(s_1, s_2) = \frac{1}{2}\text{BLEU}(s_1, s_2) \cdot \text{BLEU}(s_2, s_1) \tag{4}$$

For our experiments, we used BLEU scores of order $N = 1..4$, over n-grams formed over the sentence lemmas, and used these as features for characterizing a given pair.

4.1 Precision Focused POS Features

From past experiments with paraphrase identification over the MSR Paraphrase Corpus, we have found including POS information to be beneficial. To this capture this kind of information, we generated precision focused POS features, which measures the following between the sentences in a problem pair:

1. The overlap in POS tags.
2. The mismatch in POS tags.

We follow the formulation for POS vectors given in (Finch et al., 2005). For a given sentence pair, we identify the set of words whose lemmas were matched in both the system and reference sentences, W_{match} and those with no matches, W_{miss}. Using the directional notion of system and reference sentences from BLEU, for each word $w \in W_{match}$,

$$\text{POSMatch}(t, sys, ref) = \frac{\displaystyle\sum_{w \in W_{match}} \text{count}_t(w)}{|sys|} \tag{5}$$

where count_t is 1 if word w has the matching POS tag, and 0 otherwise. $|sys|$ is the token count of the system sentence. This is deemed to be precision-focused, as this computation is done over candidates found in the system sentence.

To generate the score for missing POS tags, we perform a similar computation,

$$\text{POSMiss}(t, sys, ref) = \frac{\displaystyle\sum_{w \in W_{miss}} \text{count}_t(w)}{|sys|} \tag{6}$$

To score the POS match and misses between a pair, we follow Formula 4 and average the scores for each POS tag, where the sentences in a given pair swap positions as the system and reference sentences.

5 Split-Bigram Features

System 2 added split-bigram features, which were derived from the ROUGE-S measure. Like bigrams, split-bigrams consist of an ordered pair of distinct tokens drawn from a source sentence. Unlike bigrams, split-bigrams allow for a number of intervening tokens to appear between the split-bigram tokens. For example, *"The cat ate fish."* would generate the following split-bigrams *the→cat*, *the→ate*, *the→fish*, *cat→ate*, *cat→fish*, and *ate→fish*. The intent of split-bigrams is to quickly capture long range dependencies, without requiring a parse of the sentence.

Similar to ROUGE-S, we used lexical overlap of the split-bigrams as an approximation of semantic similarity. As our pairs are bidirectional, we used the same framework (Formula 2) for obtaining BLEU scores to generate split-bigram overlap scores for our pairs. Here, counts are obtained over split-bigrams found in the system and reference sentences, and the order was set to 1.

For generating the skip-bigram overlap score for a pair, we used a maximum distance of three.

5.1 Skip-Bigram POS Features

In the same vein as the precision focused POS features, we used the POS tags of matched split-bigrams as features, where the frequency of the POS tags in split-bigrams, $t \rightarrow t'$, were used. Here, B_{match} represents the split-bigrams which were found in both the system and reference sentences, matched on lexical content.

$$\text{SBMatch}(t \rightarrow t', sys, ref) = \frac{\sum_{b \in B_{match}} \text{count}_{t \rightarrow t'}(b)}{|sys|} \tag{7}$$

Due to sparsity, we only considered scores from split-bigram matches between the system and reference sentences, and do not model split-bigram misses. As before, we generate scores for each split-bigram tag sequence by averaging the scores where both sentences in a pair have swapped positions. For our experiments, we only considered split-bigram POS features of up to distance 3. In our initial experiments we found split-bigram POS features helped only in the case of shorter sentence pairs, so we only generated features if both the sentences in a given pair contained ten tokens or less.

6 Experimental Setup

For all three systems, we used the Stanford CoreNLP (Toutanova et al., 2003) package to perform lemmatization and POS tagging of the input sentences. For regressors, we used LibSVM's (Chang and Lin, 2011) support vector regression capability, using radial basis kernels. Based off of tuning on the training set, we set $\gamma = 1$ and the default slack value.

Dataset	Mean	Std.Dev
MSRpar	3.322	0.9294
MSRvid	2.135	1.595
SMTeur	4.307	0.7114

Table 1: Means and standard deviations of similarity scores for each of the training datasets.

From previous experience with paraphrase identification over the MSR Paraphrase Corpus, we retained stop words in all of our experiments.

7 Dealing with Surprise Data

As the STS training data was broken into three separate datasets, each with their own distinct statistics, we developed three regressors trained individually on each of these datasets. This presented a problem when dealing with surprise datasets, whose statistics were not known.

The approach taken by Systems 1 and 2 was simply to pool together all three training datasets into a single dataset and train a single regressor on that unified model. We then applied that regressor against the two surprise datasets, OnWN and SMTnews.

Analysis of the similarity score statistics showed that they varied greatly between each of the training sets, as given in Table 1. Thus combining the datasets blindly, as with Systems 1 and 2, may prove to be a suboptimal strategy. The approach taken by System 3 was to consider the feature vectors themselves as capturing information about which dataset they were drawn from, and to use a classifier to predict that dataset. We then emit the score from the regressor trained on just that matching dataset. We used the Stanford Classifier's (Manning and Klein, 2003) multinomial logistic regression as our dataset predictor, using the feature vectors from System 2.

Five-fold cross validation over the training data showed the dataset predictor to have an overall accuracy of 91.75%.

In order to assess performance over the known datasets at test time, System 3 also applied the same strategy for the MSRpar, MSRvid, and SMTeuroparl test sets.

Sys	All	Allnorm	Mean	MSRpar	MSRvid	SMTeur	OnWN	SMTnews
1	0.7513 / 11	0.8017 / 40	0.5997 / 22	**0.6084**	0.7458	**0.4688**	**0.6315**	**0.3994**
2	0.7562 / 10	0.8111 / 24	0.5858 / 33	0.6050	**0.7939**	0.4294	0.5871	0.3366
3	0.6876 / 21	0.7812 / 54	0.4668 / 68	0.4791	0.7901	0.2159	0.3843	0.2801

Table 2: Pearson correlation of described systems against test data, by dataset. Overall measures are *All* indicates the combined Pearson, *Allnorm* the normalized variant, and *Mean* the macro average of Pearson correlations. Rank for the system in the overall measure is given after the slash.

Guess/Gold	MSRpar	MSRvid	SMTeur
MSRpar	664	7	75
MSRvid	7	737	10
SMTeur	79	6	649

Table 3: Confusion for the dataset predictor, used to predict which dataset a pair was drawn from. This was ddrawn using five-fold cross validation over the training set, with columns representing golds and guesses as rows.

Dataset	Prec	Rec	F1
MSRpar	0.8901	0.8853	0.8877
MSRvid	0.9775	0.9827	0.9801
SMTeur	0.8842	0.8842	0.8842

Table 4: Results on classifying pairs by source dataset, using five-fold cross validation over training data.

8 Results and Discussion

Results on the test data for each of the systems against the individual datasets, are given in Table 2, given in Pearson linear correlation with the gold standard scores. Overall measures for the systems are given, along with their overall ranking.

The split-bigram features in System 2 contributed primarily to performance over the MSRvid dataset, while degrading performance on the other datasets slightly. This is likely a result of increasing sparsity in the feature space, but overall this system performed well. System 3 underperformed on most datasets, asides from its performance on MSRvid. The confusion generated over five-fold cross validation over the training set is given in Table 3, and precision, recall, and F1 scores by dataset label from five-fold cross validation over the training set are given in Table 4. As these show, predictor errors lay primarily in confusing MSRpar for SMTeuroparl, and vice versa. This error was significant enough to reduce performance on both the MSRpar and SMTeuroparl test sets. This proved to be enough to reduce the scores between these two datasets.

9 Conclusion and Future Work

Our STS systems have shown that relatively simple syntax free methods can be employed to the STS task. Future avenues of investigation would be to include the use of syntactic information, in order to obtain better predicate-argument information. Syntactic information has proven useful for the paraphrase identification task over MSRpar, as demonstrated in studies such as (Das and Smith, 2009) and (Socher et al., 2011). Furthermore, a qualitative assessment of the pairs across different datasets showed relatively significant differences, which would strengthen the argument for developing features and methods specific to each dataset. Another improvement would be to develop a better dataset predictor for System 3. Also recognizing there may be ways to normalize and rescale scores across datasets so the regression models used do not have to account for differing means and standard deviations.

Finally, there are other bodies of source data that may be adapted for use with the STS task, such as the paraphrasing pairs of the Recognizing Textual Entailment challenges, human generated reference translations for machine translation evaluation, and human generated summaries used for summarization evaluations. Although these are gold decisions, at the very least they could provide a source of high similarity pairs, from which one could manufacture lower scoring variants.

Acknowledgments

The authors would like to thank the Semantic Textual Similarity organizers for all of their hard work and effort.

One of the authors was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory (AFRL) contract number FA8650-10-C-7058. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, AFRL, or the U.S. Government.

Eneko Agirre was partially funded by the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 270082 (PATHS project) and the Ministry of Economy under grant TIN2009-14715-C04-01 (KNOW2 project)

References

Eneko Agirre, Montse Cuadros, German Rigau, and Aitor Soroa. 2010. Exploring knowledge bases for similarity. In *Proceedings of the International Conference on Language Resources and Evaluation 2010*.

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)*.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27.

Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *In Proceedings of the Joint Conference of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing(ACL 2009)*, pages 468–476, Singapore.

Christine Fellbaum. 1998. *WordNet - An Electronic Lexical Database*. MIT Press.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloqium*.

Andrew Finch, Young-Sook Hwang, and Eiichio Sumita. 2005. Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*, pages 17–24, Jeju Island, South Korea.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Evgeniy Gabrilovich and Shaul Markovitch. 2009. Wikipedia-based semantic interpretation. *Journal of Artificial Intelligence Research*, 34:443–498.

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *WWW '02*, pages 517–526, New York, NY, USA. ACM.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. pages 74–81, Barcelona, Spain, jul. Association for Computational Linguistics.

Christopher Manning and Dan Klein. 2003. Optimization, maxent models, and conditional estimation without magic. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Tutorials - Volume 5*, NAACL-Tutorials '03, pages 8–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November. Previous number = SIDL-WP-1999-0120.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (Intelligent Systems Demonstrations)*, pages 1024–1025, San Jose, CA, July.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems 24*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.