UOW: Semantically Informed Text Similarity

Miguel Rios and Wilker Aziz
Research Group in Computational Linguistics
University of Wolverhampton
Stafford Street, Wolverhampton,
WV1 1SB, UK
{M.Rios, W.Aziz}@wlv.ac.uk

Lucia Specia
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello,
Sheffield, S1 4DP, UK
L.Specia@sheffield.ac.uk

Abstract

The UOW submissions to the Semantic Textual Similarity task at SemEval-2012 use a supervised machine learning algorithm along with features based on lexical, syntactic and semantic similarity metrics to predict the semantic equivalence between a pair of sentences. The lexical metrics are based on word-overlap. A shallow syntactic metric is based on the overlap of base-phrase labels. The semantically informed metrics are based on the preservation of named entities and on the alignment of verb predicates and the overlap of argument roles using inexact matching. Our submissions outperformed the official baseline, with our best system ranked above average, but the contribution of the semantic metrics was not conclusive.

1 Introduction

We describe the UOW submissions to the Semantic Textual Similarity (STS) task at SemEval-2012. Our systems are based on combining similarity scores as features using a regression algorithm to predict the degree of semantic equivalence between a pair of sentences. We train the regression algorithm with different classes of similarity metrics: i) lexical, ii) syntactic and iii) semantic. The lexical similarity metrics are: i) cosine similarity using a bag-of-words representation, and ii) precision, recall and F-measure of content words. The syntactic metric computes BLEU (Papineni et al., 2002), a machine translation evaluation metric, over a labels of base-phrases (chunks). Two semantic metrics are used: a metric based on the preservation of Named Entities and TINE (Rios et al., 2011). Named entities are matched by type and content: while the type has to match exactly, the content is compared with the assistance of a distributional thesaurus. TINE is a metric proposed to measure adequacy in machine translation and favors similar semantic frames. TINE attempts to align verb predicates, assuming a one-to-one correspondence between semantic roles, and considering ontologies for inexact alignment. The surface realization of the arguments is compared using a distributional thesaurus and the cosine similarity metric. Finally, we use METEOR (Denkowski and Lavie, 2010), also a common metric for machine translation evaluation, that also computes inexact word overlap as at way of measuring the impact of our semantic metrics.

The lexical and syntactic metrics complement the semantic metrics in dealing with the phenomena observed in the task's dataset. For instance, from the MSRvid dataset:

S1 *Two men are playing football.*

S2 *Two men are practicing football.*

In this case, as typical of paraphrasing, the situation and participants are the same while the surface realization differs, but *playing* can be considered similar to *practicing*. From the SMT-eur dataset:

S3 *The Council of Europe, along with the Court of Human Rights, has a wealth of experience of such forms of supervision, and we can build on these.*

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 673–678,
Montréal, Canada, June 7-8, 2012.

S4 *Just as the European Court of Human Rights, the Council of Europe has also considerable experience with regard to these forms of control; we can take as a basis.*

Similarly, here although with different realizations, the *Court of Human Rights* and the *European Court of Human Rights* represent the same entity.

Semantic metrics based on predicate-argument structure can play a role in cases when different realization have similar semantic roles:

S5 *The right of a government arbitrarily to set aside its own constitution is the defining characteristic of a tyranny.*

S6 *The right for a government to draw aside its constitution arbitrarily is the definition characteristic of a tyranny.*

In this work we attempt to exploit the fact that superficial variations such the ones in these examples should still render very similarity scores.

In Section 2 we describe the similarity metrics in more detail. In Section 3 we show the results of our three systems. In Section 4 we discuss these results and in Section 5 we present some conclusions.

2 Similarity Metrics

The metrics used in this work are as follows:

2.1 Lexical metrics

All our lexical metrics use the same surface representation: words. However, the cosine metric uses bag-of-words, while all the other metrics use only content words. We thus first represent the sentences as bag-of-words. For example, given the pair of sentences S7 and S8:

S7 *A man is riding a bicycle.*

S8 *A man is riding a bike.*

the bag-of-words are S7 = {*A, man, is, riding, a, bicycle,.*} and S8 = {*A, man, is, riding, a, bike, .*}, and the bag-of-content-words are S7 = {*man, riding, bicycle*} and S8 = {*man, riding, bike*}.

We compute similarity scores using the following metrics between a pair of sentences A and B: cosine distance (Equation 1), precision (Equation 2), recall (Equation 3) and F-measure (Equation 4).

$$cosine(A,B) = \frac{|A \bigcap B|}{\sqrt{|A| \times |B|}} \quad (1)$$

$$precision(A,B) = \frac{|A \bigcap B|}{|B|} \quad (2)$$

$$recall(A,B) = \frac{|A \bigcap B|}{|A|} \quad (3)$$

$$F(A,B) = 2 \cdot \frac{precision(A,B) \cdot recall(A,B)}{precision(A,B) + recall(A,B)} \quad (4)$$

2.2 BLEU over base-phrases

The BLEU metric is used for the automatic evaluation of Machine Translation. The metric computes the precision of exact matching of n-grams between a hypothesis and reference translations. This simple procedure has limitations such as: the matching of non-content words mixed with the counts of content words affects in a perfect matching that can happen even if the order of sequences of n-grams in the hypothesis and reference translation are very different, changing completely the meaning of the translation. To account for similarity in word order we use BLEU over base-phrase labels instead of words, leaving the lexical matching for other lexical and semantic metrics. We compute the matchings of 1-4-grams of base-phrase labels. This metric favors similar syntactic order.

2.3 Named Entities metric

The goal of the metric is to deal with synonym entities. First, named entities are grouped by class (e.g. Organization), and then the content of the named entities within the same classes is compared through cosine similarity. If the surface realization is different, we retrieve words that share the same context with the named entity using Dekang Lin's distributional thesaurus (Lin, 1998). Therefore, the cosine similarity will have more information than just the named entities themselves. For example, from the sentence pair S9 and S10:

S9 *Companies include IBM Corp. ...*

S10 *Companies include International Business Machines ...*

The entity from S9: *IBM Corp.* and the entity from S10: *International Business Machines* have the same tag *Organization*. The metric groups them and adds words from the thesaurus resulting in the following bag-of-words. S9: {*IBM Corp.,... Microsoft, Intel, Sun Microsystems, Motorola/Motorola, Hewlett-Packard/Hewlett-Packard, Novell, Apple Computer...*} and S10: {*International Business Machines,... Apple Computer, Yahoo, Microsoft, Alcoa...*}. The metric then computes the cosine similarity between this expanded pair of bag-of-words.

2.4 METEOR

This metric is also a lexical metric based on unigram matching between two sentences. However, matches can be exact, using stems, synonyms, or paraphrases of unigrams. The synonym matching is computed using WordNet (Fellbaum, 1998) and the paraphrase matching is computed using paraphrase tables (Callison-Burch et al., 2010). The structure of the sentences is not not directly considered, but similar word orders are rewarded through higher scores for the matching of longer fragments.

2.5 Semantic Role Label metric

Rios et al. (2011) propose TINE, an automatic metric based on the use semantic roles to align predicates and their respective arguments in a pair of sentences. The metric complements lexical matching with a shallow semantic component to better address adequacy in machine translation evaluation. The main contribution of such a metric is to provide a more flexible way of measuring the overlap between shallow semantic representations (semantic role labels) that considers both the semantic structure of the sentence and the content of the semantic components.

This metric allows to match synonym predicates by using verb ontologies such as VerbNet (Schuler, 2006) and VerbOcean (Chklovski and Pantel, 2004) and distributional semantics similarity metrics, such as Dekang Lin's thesaurus (Lin, 1998), where previous semantic metrics only perform exact match of predicate structures and arguments. For example, in VerbNet the verbs *spook* and *terrify* share the same class *amuse-31.1*, and in VerbOcean the verb *dress* is related to the verb *wear*, so these are considered matches in TINE.

The main sources of errors in this metric are the matching of unrelated verbs and the lack of coverage of the ontologies. For example, for S11 and S12, *remain* and *say* are (incorrectly) related as given by VerbOcean.

S11 *If snow falls on the slopes this week, Christmas will sell out too, says Schiefert.*

S12 *If the roads remain snowfall during the week, the dates of Christmas will dry up, said Schiefert.*

For this work the matching of unrelated verbs is a particularly crucial issue, since the sentences to be compared are not necessarily similar, as it is the general case in machine translation. We have thus modified the metric with a preliminary optimization step which aligns the verb predicates by measuring two degrees of similarity: i) how similar their arguments are, and ii) how related the predicates' realizations are. Both scores are combined as shown in Equation 5 to score the similarity between the two predicates (A_v, B_v) from a pair of sentences (A, B).

$$\begin{aligned}\mathrm{sim}(\mathrm{A_v}, \mathrm{B_v}) = (w_{lex} \times lexScore(A_v, B_v)) \\ +(w_{arg} \times argScore(A_{arg}, B_{arg}))\end{aligned} \quad (5)$$

where w_{lex} and w_{arg} are the weights for each component, $argScore(A_{arg}, B_{arg})$ is the similarity, which is computed as in Equation 7, of the arguments between the predicates being compared and $lexScore(A_v, B_v)$ is the similarity score extracted from the Dekang Lin's thesaurus between the predicates being compared. The Dekang Lin's thesaurus is an automatically built thesaurus, and for each word it has an entry with the most similar words and their similarity scores. If the verbs are related in the thesaurus we use their similarity score as $lexScore$ otherwise $lexScore = 0$. The pair of predicates with the maximum sim score is aligned. The alignment is an optimization problem where predicates are aligned 1-1: we search for all 1-1 alignments that lead to the maximum average sim for the pair of sentences. For example, S13 and S14 have the following list of predicates: S13 = {loaded, rose, ending}

and S14 = {laced, climbed}. The metric compares each pair of predicates and it aligns the predicates *rose* and *climbed* because they are related in the thesaurus with a similarity score $lexScore = 0.796$ and a $argScore = 0.185$ given that the weights are set to 0.5 and sum up to 1 the predicates reach the maximum $\text{sim} = 0.429$ score. The output of this step results in a set of aligned verbs between a pair of sentences.

S13 *The tech - loaded Nasdaq composite rose 0 points to 0 , ending at its highest level for 0 months.*

S14 *The technology - laced Nasdaq Composite Index IXIC climbed 0 points , or 0 percent , to 0.*

The SRL similarity metric $semanticRole$ between two sentences A and B is then defined as:

$$semanticRole(A, B) = \frac{\sum_{v \in V} verbScore(A_v, B_v)}{|V_B|} \quad (6)$$

The $verbScore$ in Equation 6 is computed over the set of aligned predicates from the previous optimization step and for each aligned predicate the argument similarity is computed by Equation 7.

$$verbScore(A_v, B_v) = \frac{\sum_{arg \in Arg_A \cap Arg_B} argScore(A_{arg}, B_{arg})}{|Arg_B|} \quad (7)$$

In Equation 6, *V* is the set of verbs aligned between the two sentences A and B, and $|V_B|$ is the number of verbs in one of the sentences.[1] The similarity between the arguments of a verb pair (A_v, B_v) in *V* is measured as defined in Equation 7, where Arg_A and Arg_B are the sets of labeled arguments of the first and the second sentences and $|Arg_B|$ is the number of arguments of the verb in B.[2] The $argScore(A_{arg}, B_{arg})$ computation is based on the cosine similarity as in Equation 1. We treat the tokens in the argument as a bag-of-words.

[1]This is inherited from the use of the metric focusing on recall in machine translation, where the B is the reference translation. In this work a better approach could be to compute this metric twice, in both directions.

[2]Again, from the analogy of a recall metric for machine translation.

3 Experiments and Results

We use the following state-of-the-art tools to preprocess the data for feature extraction: i) TreeTagger[3] for lemmas and ii) SENNA (Collobert et al., 2011)[4] for Part-of-Speech tagging, Chunking, Named Entity Recognition and Semantic Role Labeling. SENNA has been reported to achieve an F-measure of 75.79% for tagging semantic roles on the CoNLL-2005 [2] benchmark. The final feature set includes:

- Lexical metrics
  - Cosine metric over bag-of-words
  - Precision over content words
  - Recall over content words
  - F-measure over content words
- BLEU metric over chunks
- METEOR metric over words (with stems, synonyms and paraphrases)
- Named Entity metric
- Semantic Role Labeling metric

The Machine Learning algorithm used for regression is the LIBSVM[5] Support Vector Machine (SVM) implementation using the radial basis kernel function. We used a simple genetic algorithm (Back et al., 1999) to tune the parameters of the SVM. The configuration of the genetic algorithm is as follows:

- Fitness function: minimize the mean squared error found by cross-validation
- Chromosome: real numbers for SVM parameters γ, *cost* and ϵ
- Number of individuals: 80
- Number of generations: 100
- Selection method: roulette
- Crossover probability: 0.9

[3]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

[4]http://ml.nec-labs.com/senna/

[2]http://www.lsi.upc.edu/ srlconll/

[5]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

- Mutation probability: 0.01

We submitted three system runs, each is a variation of the above feature set. For the official submission we used the systems with optimized SVM parameters. We trained SVM models with each of the following task datasets: MSRpar, MSRvid, SMT-eur and the combination of MSRpar+MSRvid. For each test dataset we applied their respective training models, except for the new test sets, not covered by any training set: for On-WN we used the combination MSRpar+MSRvid, and for SMT-news we used SMT-eur.

Tables 1 to 3 focus on the Pearson correlation of our three systems/runs for individual datasets of the predicted scores against human annotation, compared against the official baseline, which uses a simple word overlap metric. Table 4 shows the average results over all five datasets, where ALL stands for the Pearson correlation with the gold standard for the five dataset, Rank is the absolute rank among all submissions, ALLnrm is the Pearson correlation when each dataset is fitted to the gold standard using least squares, RankNrm is the corresponding rank and Mean is the weighted mean across the five datasets, where the weight depends on the number of sentence pairs in the dataset.

3.1 Run 1: All except SRL features

Our first run uses the lexical, BLEU, METEOR and Named Entities features, without the SRL feature. Table 1 shows the results over the test set, where Run 1-A is the version without SVM parameter optimization and Run 1-B are the official results with optimized parameters for SVM.

Task	Run 1-A	Run 1-B	Baseline
MSRpar	0.455	0.455	0.433
MSRvid	0.706	0.362	0.300
SMT-eur	0.461	0.307	0.454
On-WN	0.514	0.281	0.586
SMT-news	0.386	0.208	0.390

Table 1: Results for Run 1 using lexical, chunking, named entities and METEOR as features. A is the non-optimized version, B are the official results

3.2 Run 2: SRL feature

In this run we use only the SRL feature in order to analyze whether this feature on its own could be sufficient or lexical and other simpler features are important. Table 2 shows the results over the test set without parameter optimization (Run 2-A) and the official results with optimized parameters for SVM (Run 2-B).

Task	Run 2-A	Run 2-B	Baseline
MSRpar	0.335	0.300	0.433
MSRvid	0.264	0.291	0.300
SMT-eur	0.264	0.161	0.454
On-WN	0.281	0.257	0.586
SMT-news	0.189	0.221	0.390

Table 2: Results for Run 2 using the SRL feature only. A is the non-optimized version, B are the official results

3.3 Run 3: All features

In the last run we use all features. Table 3 shows the results over the test set without parameter optimization (Run 3-A) and the official results with optimized parameters for SVM (Run 3-B).

Task	Run 3-A	Run 3-B	Baseline
MSRpar	0.472	0.353	0.433
MSRvid	0.705	0.572	0.300
SMT-eur	0.471	0.307	0.454
On-WN	0.511	0.264	0.586
SMT-news	0.410	0.116	0.390

Table 3: Results for Run 3 using all features. A is the non-optimized version, B are the official results

4 Discussion

Table 4 shows the ranking and normalized official scores of our submissions compared against the baseline. Our submissions outperform the official baseline but significantly underperform the top systems in the shared task. The best system (Run 1) achieved an above average ranking, but disappointingly the performance of our most complete system (Run 3) using the semantic metric is poorer. Surprisingly, the results of the non-optimized versions outperform the optimized versions used in our official submission. One possible reason for that is the overfitting of the optimized models to the training sets.

Run 1 and Run 3 have very similar results: the overall correlation between all datasets of these two systems is 0.98. One of the reasons for these results is that the SRL metric is compromised by the length

System	ALL	Rank	ALLnrm	RankNrm	Mean	RankMean
Run 1	0.640	36	0.719	71	0.382	80
Run 2	0.536	59	0.629	88	0.257	88
Run 3	0.598	49	0.696	82	0.347	84
Baseline	0.311	87	0.673	85	0.436	70

Table 4: Official results and ranking over the test set for Runs 1-3 with SVM parameters optimized

of the sentences. In the MSRvid dataset, where the sentences are simple such as "*Someone is drawing*", resulting in a good semantic parsing, a high performance for this metric is achieved. However, in the SMT datasets, sentences are much longer (and often ungrammatical, since they are produced by a machine translation system) and the performance of the metric drops. In addition, the SRL metric makes mistakes such as judging as highly similar sentences such as "*A man is peeling a potato*" and "*A man is slicing a potato*", where the arguments are the same but the situations are different.

5 Conclusions

We have presented our systems based on similarity scores as features to train a regression algorithm to predict the semantic similarity between a pair of sentences. Our official submissions outperform the baseline method, but have lower performance than most participants, and a simpler version of the systems without any parameter optimization proved more robust. Disappointingly, our main contribution, the addition of a metric based on Semantic Role Labels shows no improvement as compared to simpler metrics.

Acknowledgments

This work was supported by the Mexican National Council for Science and Technology (CONACYT), scholarship reference 309261.

References

Thomas Back, David B. Fogel, and Zbigniew Michalewicz, editors. 1999. *Evolutionary Computation 1, Basic Algorithms and Operators*. IOP Publishing Ltd., Bristol, UK, 1st edition.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (almost) from Scratch.

Michael Denkowski and Alon Lavie. 2010. Meteor-next and the meteor paraphrase tables: Improved evaluation support for five target languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 339–342, July.

Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. Cambridge, MA ; London, May.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA.

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.

Miguel Rios, Wilker Aziz, and Lucia Specia. 2011. Tine: A metric to assess mt adequacy. Proceedings of the Sixth Workshop on Statistical Machine Translation.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.