UoW: NLP Techniques Developed at the University of Wolverhampton for Semantic Similarity and Textual Entailment

Rohit Gupta, Hanna Béchara, Ismail El Maarouf and **Constantin Orăsan**
Research Group in Computational Linguistics,
Research Institute of Information and Language Processing,
University of Wolverhampton, UK
{R.Gupta, Hanna.Bechara, I.El-Maarouf, C.Orasan}@wlv.ac.uk

Abstract

This paper presents the system submitted by University of Wolverhampton for SemEval-2014 task 1. We proposed a machine learning approach which is based on features extracted using Typed Dependencies, Paraphrasing, Machine Translation evaluation metrics, Quality Estimation metrics and Corpus Pattern Analysis. Our system performed satisfactorily and obtained 0.711 Pearson correlation for the semantic relatedness task and 78.52% accuracy for the textual entailment task.

1 Introduction

The SemEval task 1 (Marelli et al., 2014a) involves two subtasks: predicting the degree of relatedness between two sentences and detecting the entailment relation holding between them. The task uses SICK dataset (Marelli et al., 2014b), consisting of 10000 pairs, each annotated with relatedness in meaning and entailment relationship holding between them. Similarity measures between sentences are required in a wide variety of NLP applications. In applications like Information Retrieval (IR), measuring similarity is a vital step in order to determine the best result for a related query. Other applications such as Paraphrasing and Translation Memory (TM) rely on similarity measures to weight results. However, computing semantic similarity between sentences is a complex and difficult task, due to the fact that the same meaning can be expressed in a variety of ways. For this reason it is necessary to have more than a surface-form comparison.

We present a method based on machine learning which exploits available NLP technology. Our approach relies on features inspired by deep semantics (such as parsing and paraphrasing), machine translation quality estimation, machine translation evaluation and Corpus Pattern Analysis (CPA[1]).

We use the same features to measure both semantic relatedness and textual entailment. Our hypothesis is that each feature covers a particular aspect of implicit similarity and entailment information contained within the pair of sentences. Training is performed in a regression framework for semantic relatedness and in a classification framework for textual entailment.

The remainder of the paper is structured as follows. In Section 2, we review the work related to our study and the existing NLP technologies used to measure sentence similarity. In Sections 3 and 4, we describe our approach and the similarity measures we used. In Section 5, we present the results and an analysis of our runs based on the test and training data provided by the SemEval-2014 task. Finally, our work is summed up in Section 6 with perspectives for future work we would like to explore.

2 Related Work

The areas of semantic relatedness and entailment have received extensive interest from the research community in the last decade. Earlier work in relatedness (Banerjee and Pedersen, 2003; Li et al., 2006) exploited WordNet in various ways to extract the semantic relatedness. Banerjee and Pedersen (2003) presented a measure using extended gloss overlap. This measure takes two WordNet synsets as input and uses the overlap of their WordNet glosses to compute their degree of semantic relatedness. Li et al. (2006) presented a semantic similarity metric based on the semantic similarity of words in a sentence. Recently, Wang and Cer (2012) presented an ap-

[1] http://pdev.org.uk

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 785–789,
Dublin, Ireland, August 23-24, 2014.

proach that uses probabilistic edit-distance to measure semantic similarity. The approach uses probabilistic finite state and pushdown automata to model weighted edit-distance where state transitions correspond to edit-operations. In some aspects, our work is similar to Bär et al. (2012), who presented an approach which combines various text similarity measures using a log-linear regression model.

Entailment has been modelled using various approaches. The main approaches are based on logic inferencing (Moldovan et al., 2003), machine learning (Hickl et al., 2006; Castillo, 2010) and tree edit-distance (Kouylekov and Magnini, 2005). Most of the recent approaches employ various syntactic or tree edit models (Heilman and Smith, 2010; Mai et al., 2011; Rios and Gelbukh, 2012; Alabbas and Ramsay, 2013). Recently, Alabbas and Ramsay (2013) presented a modified tree edit distance approach, which extends tree edit distance to the level of subtrees. The approach extends Zhang-Shasha's algorithm (Zhang and Shasha, 1989).

3 Features

Our system uses the same 31 features for both subtasks. This section explains them and the code which implements most of them can be found on GitHub[2].

3.1 Language Technology Features

We used existing language processing tools to extract features. Stanford CoreNLP[3] toolkit provides lemma, parts of speech (POS), named entities, dependencies relations of words in each sentence.

We calculated Jaccard similarity on surface form, lemma, dependencies relations, POS and named entities to get the feature values. The Jaccard similarity computes sentence similarity by dividing the overlap of words on the total number of words of both sentences.

$$Sim(s1, s2) = \frac{|s1 \cap s2|}{|s1 \cup s2|} \quad (1)$$

where in equation (1), $Sim(s1, s2)$ is the Jaccard similarity between sets of words $s1$ and $s2$.

We used the same toolkit to identify coreference relations and determine clusters of coreferential entities. The coreference feature value was calculated using clusters of coreferential entities. The intuition is that sentences containing coreferential entities should have some semantic relatedness. In order to extract clusters of coreferential entities, the pair of sentences was treated as a document. The coreference feature value using these clusters was calculated as follows:

$$Value_{coref} = \frac{CC}{TC} \quad (2)$$

where CC is the number of clusters formed by the participation of entities (at least one entity from each sentence of the pair) in both sentences and TC is the total number of clusters.

We calculated two separate feature values for dependency relations: the first feature concatenated the words involved in a dependency relation and the second used grammatical relation tags. For example, for the sentence pair "the kids are playing outdoors" and "the students are playing outdoors" the Jaccard similarity is calculated based on concatenated words "kids::the, playing::kids, playing::are, ROOT::playing, playing::outdoors" and "students::the, playing::students, playing::are, ROOT::playing, playing::outdoors" to get the value for the first feature and "det, nsubj, aux, root, dobj" and "det, nsubj, aux, root, dobj" to get the value for the second feature.

These language technology features try to capture the token based similarity and grammatical similarity between a pair of sentences.

3.2 Paraphrasing Features

We used the PPDB paraphrase database (Ganitkevitch et al., 2013) to get the paraphrases. We used lexical and phrasal paraphrases of "L" size. For each sentence of the pair, we created two sets of bags of n-grams ($1 \leq n \leq$ length of the sentence). We extended each set with paraphrases for each n-gram available from paraphrase database. We then calculated the Jaccard similarity (see Section 3.1) between these extended bag of n-grams to get the feature value. This feature capture the cases where one sentence is a paraphrase of the other.

3.3 Negation Feature

Our system does not attempt to model similarity with negation, but since negation is an important feature for contradiction in textual entailment, we designed a non-similarity feature. The system checks for the presence of a negation word such as 'no', 'never' and 'not' in the pair of sentences and

[2]https://github.com/rohitguptacs/wlvsimilarity
[3]http://nlp.stanford.edu/software/corenlp.shtml

returns "1" ("0" otherwise) if both or none of the sentences contain any of these words.

3.4 Machine Translation Quality Estimation Features

Seventeen of the features consist of Machine Translation Quality Estimation (QE) features, based on the work of (Specia et al., 2009) and used as a baseline in recent QE tasks (such as (Callison-Burch et al., 2012)). We extracted these features by treating the first set of sentences as the Machine Translation (MT) "source", and the second set of sentences as the MT "target". In Machine Translation, these features are used to access the quality of MT "target". The QE features include shallow surface features such as the number of punctuation marks, the average length of words, the number of words. Furthermore, these features include n-gram frequencies and language model probabilities. A full list of the QE features is provided in the documentation of the QE system[4] (Specia et al., 2009).

QE features relate to well-formedness and syntax, and are not usually used to compute semantic relatedness between sentences. We have used them in the hope that the surface features at least will show us some structural similarity between sentences.

3.5 Machine Translation Evaluation Features

Additionally, we used BLEU (Papineni et al., 2002), a very popular machine translation evaluation metric, as a feature. BLEU is based on n-gram counts. It is meant to capture the similarity between translated text and references for machine translation evaluation. The BLEU score over surface, lemma and POS was calculated to get three feature values. In a pair of sentences, one side was treated as a translation and another as a reference. We applied it at the sentence level to capture the similarity between two sentences.

3.6 Corpus Pattern Analysis Features

Corpus Pattern Analysis (CPA) (Hanks, 2013) is a procedure in corpus linguistics that associates word meaning with word use by means of semantic patterns. CPA is a new technique for mapping meaning onto words in text. It is currently being used to build a "Pattern Dictionary of English Verbs"(PDEV[5]). It is based on the Theory of Norms and Exploitations (Hanks, 2013).

There are two features extracted from PDEV. They both make use of a derived resource called the CPA network (Bradbury and El Maarouf, 2013). The CPA network links verbs according to similar semantic patterns (e.g. both 'pour' and 'trickle' share an intransitive use where the subject is "liquid").

The first feature value compares the main verbs in both sentences. When both verbs share a pattern, the system returns a value of "1" (otherwise "0"). The second feature extends the CPA network to compute the probability of a PDEV pattern, given a word. This probability is computed over the portion of the British National Corpus which is manually tagged with PDEV patterns. The probability of a pattern given each word of a sentence of the dataset is obtained by the product of those probabilities. The feature value is the (normalised) number of common patterns from the three most probable patterns in each sentence. These features try to capture similarity based on semantic patterns.

4 Predicting Through Machine Learning

4.1 Model Description

We used a support vector machine in order to build a regression model to predict semantic relatedness and a classification model to predict textual entailment. For the actual implementation we used LibSVM[6] (Chang and Lin, 2011).

We used a regression model for the relatedness task that estimates a continuous score between 1 and 5 for each sentence. For the entailment task, we trained a classification model which assigns one of three different labels (ENTAILMENT, CONTRADICTION, NEUTRAL) to each sentence pair. We trained both systems on the 4500 sentence training set, augmented with the 500 sentence trial data. The values of C and γ have been optimised through a grid-search which uses a 5-fold cross-validation method.

The RBF kernel proved to be the best for both tasks.

5 Results and Analysis

We submitted 4 runs of our system (Run-1 to Run-4). Run-1 was submitted as primary run. Run-2, Run-3 and Run-4 systems were identical except

[4]https://github.com/lspecia/quest

[5]http://pdev.org.uk

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

	Run-1	Run-2	Run-3	Run-4
C	8	8	2	2
γ	0.0441	0.0441	0.125	0.125
Pearson	0.7111	0.7166	0.6968	0.6975

Table 1: Results: Relatedness.

	Run-1	Run-2	Run-3	Run-4
C	16	16	8	8
γ	0.0625	0.0625	0.5	0.5
Accuracy	78.526	78.526	78.343	78.343

Table 2: Results: Entailment.

for some parameter differences for SVM training and the replacement of the values which were outside the boundaries (1-5). If relatedness values predicted by the system were less than 1 or greater than 5, these values were replaced by 1 and 5 respectively for Run-1, Run-2 and Run-4 and 1.5 and 4.5 respectively for Run-3. Our primary run also used one extra feature for relatedness, which was obtained by considering entailment judgement as a feature. Our hypothesis was that entailment judgement may help in measuring relatedness. In the actual test this feature was not helpful and we obtained Pearson correlation of 0.711 for the primary run, compared to 0.716 for Run-2. The details of runs are given in Table 1 and 2.

After training both models, we ran a feature selection algorithm to determine which features yielded the highest accuracy, and therefore had the highest impact on our system. Perhaps unsurprisingly, the QE features were not very useful in predicting semantic similarity or entailment. However, despite their focus on fluency rather than semantic correctness, the QE features still managed to contribute to some improvements in the textual entailment task (increasing accuracy by 1%), and the semantic relatedness task (0.027 increase in Pearson correlation).

In the entailment (classification) task, the strongest feature proved to be the negation feature with 70% accuracy (on the training set) when training on this feature only. This suggests that some measure of negation is crucial in determining whether a sentence contradicts or entails another sentence. Other strong features were lemma, paraphrasing and dependencies.

In the relatedness (regression) task, the lemma, surface, paraphrasing, dependencies, PDEV features were the strongest contributors to accuracy.

6 Conclusion and Future Work

We have presented an efficient approach to calculate semantic relatedness and textual entailment. One noticeable point of our approach is that we have used the same features for both tasks and our system performed well in each of these tasks. Therefore, our system captures reasonably good models to compute semantic relatedness and textual entailment.

In the future we would like to explore more features and particularly those based on tree edit distance, WordNet and PDEV. Our intuition suggests that tree edit distance seems to be more helpful for entailment, whereas WordNet and PDEV seem to be more helpful for similarity measurement. Additionally, we would like to combine our techniques for measuring relatedness and entailment with MT evaluation techniques. We would further like to apply these techniques cross-lingually, moving into other areas like machine translation evaluation and quality estimation.

Acknowledgement

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement no. 317471 and partly supported by an AHRC grant "Disambiguating Verbs by Collocation project, AH/J005940/1, 2012-2015".

References

Maytham Alabbas and Allan Ramsay. 2013. Natural language inference for Arabic using extended tree edit distance with subtrees. *Journal of Artificial Intelligence Research*, 48:1–22.

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, volume 3, pages 805–810.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *First Joint Conference on*

Lexical and Computational Semantics, Association for Computational Linguistics, pages 435–440.

Jane Bradbury and Ismaïl El Maarouf. 2013. An empirical classification of verbs based on Semantic Types: the case of the 'poison' verbs. In *Proceedings of the Joint Symposium on Semantic Processing. Textual Inference and Structures in Corpora*, pages 70–74.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia, editors. 2012. *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Montréal, Canada, June.

Julio J. Castillo. 2010. Recognizing textual entailment: experiments with machine learning algorithms and RTE corpora. *Special issue: Natural Language Processings and its Applications, Research in Computing Science*, 46:155–164.

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.

Juri Ganitkevitch, Van Durme Benjamin, and Chris Callison-Burch. 2013. Ppdb: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia.

Patrick Hanks. 2013. *Lexical Analysis: Norms and Exploitations*. Mit Press.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *The 2010 Annual Conference of the North American Chapter of the ACL*, number June, pages 1011–1019.

Andrew Hickl, Jeremy Bensley, John Williams, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing textual entailment with LCC's GROUNDHOG system. In *Proceedings of the Second PASCAL Challenges Workshop*.

Milen Kouylekov and Bernardo Magnini. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pages 17–20.

Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8):1138–1150.

Zhewei Mai, Y Zhang, and Donghong Ji. 2011. Recognizing text entailment via syntactic tree matching. In *Proceedings of NTCIR-9 Workshop Meeting*, pages 361–364, Tokyo, Japan.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014a. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014)*.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014b. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014*.

Dan Moldovan, Christine Clark, Sanda Harabagiu, and Steve Maiorano. 2003. COGEX : A Logic Prover for Question Answering. In *Proceedings of HLT-NAACL*, number June, pages 87–93.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the ACL*, pages 311–318.

Miguel Rios and Alexander Gelbukh. 2012. Recognizing Textual Entailment with a Semantic Edit Distance Metric. In *11th Mexican International Conference on Artificial Intelligence*, pages 15–20. IEEE.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Mengqiu Wang and Daniel Cer. 2012. Stanford: probabilistic edit distance metrics for STS. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pages 648–654.

Kaizhong Zhang and Dennis Shasha. 1989. Simple Fast Algorithms for the Editing Distance between Trees and Related Problems. *SIAM Journal on Computing*, 18(6):1245–1262.