ROB: Using Semantic Meaning to Recognize Paraphrases

Rob van der Goot
University of Groningen
r.van.der.goot@rug.nl

Gertjan van Noord
University of Groningen
g.j.m.van.noord@rug.nl

Abstract

Paraphrase recognition is the task of identifying whether two pieces of natural language represent similar meanings. This paper describes a system participating in the shared task 1 of SemEval 2015, which is about paraphrase detection and semantic similarity in twitter. Our approach is to exploit semantically meaningful features to detect paraphrases. An existing state-of-the-art model for predicting semantic similarity is adapted to this task.

A wide variety of features is used, ranging from different types of models, to lexical overlap and synset overlap. A maximum entropy classifier is then trained on these features. In addition to the detection of paraphrases, a similarity score is also predicted, using the probabilities of the classifier. To improve the results, normalization is used as preprocessing step.

Our final system achieves a F1 score of 0.620 (10th out of 18 teams), and a Pearson correlation of 0.515 (6th out of 13 teams).

1 Introduction

A good paraphrase detection system can be useful in many natural language processing tasks, like searching, translating or summarization. For clean texts, F1 scores as high as 0.84 have been reported on paraphrase detection (Madnani et al., 2012).

However, previous research focused almost solely on clean text. Thanks to the Twitter Paraphrase Corpus (Xu et al., 2014), this has now changed. Carrying out this task on noisy texts is a new challenge. The abundant availability of social media data and the high redundancy that naturally exists in this data makes this task highly relevant (Zanzotto et al., 2011).

Our approach is based on the model described by Bjerva et al. (2014). This model has proved to achieve state-of-the-art results at predicting semantic similarity (Marelli et al., 2014). It is based on overlaps of semantically meaningful properties of sentences. A random forest regression model (Breiman, 2001) combines these features to predict a semantic similarity score. We rely heavily on the assumption that semantically meaningful features can also be used to identify paraphrases.

The features of the existing system are also used in the new system. However, the old system used a regression model, while the new task demands class-based output. Hence, the machine learning model model is changed to a maximum entropy model.

2 Data

The Twitter Paraphrase Corpus consists of two distinct parts, the training data differs significantly from the test data.

The 17,790 tweet pairs for training are collected between April 24th and May 3rd, 2014. These tweets are selected based on the trending topics of that period. Annotation of the training data is done by human annotators from Amazon Mechanical Turk. Every sentence pair is annotated by 5 different annotators, resulting in a score of 0-5. Based on this score we create a binary paraphrase judgement. If 0, 1 or 2 annotators judged positively, we treat the sentence pair as not being a paraphrase, for 3, 4 or 5 positive judgements we treat the sentence

pair as a paraphrase.

The test data is collected between May 13th and June 10th, and is thus based on different trending topics. This assures the integrity of the evaluation. In contrast to the training data, this data is annotated by an expert similarity rating on a 5-point Likert scale (Likert, 1932), to mimic the training data. Sentence pairs with a similarity score of 0, 1 and 2 are considered non-paraphrases, and sentence pairs with scores of 4 and 5 are considered paraphrases. The one uncertain category (similarity score of 3) is discarded in the evaluation.

Using this data, we end up with two different types of gold data per sentence pair. Firstly, we have the binary gold data that indicates if a sentence pair is a paraphrase. Secondly, we have the raw annotations that can be used as a similarity score. These annotations are normalized by dividing them by their maximum score (5), so we end up with $\langle 0.0, 0.2, 0.4, 0.6, 0.8, 1.0 \rangle$ as possible similarity scores.

The tweets in the corpus are already tokenized using TweetMotif (O'Connor et al., 2010). Additionally, Part Of Speech (POS) tags are provided by a tagger that is adapted to twitter (Derczynski et al., 2013). Named entity tags are also obtained from an adapted tagger (Ritter et al., 2011).

3 Method

The model is based on a state-of-the-art semantic similarity prediction model (Bjerva et al., 2014). It is mainly based on overlap features extracted from different parsers, but also includes synset overlap, and a Compositional Distributional Semantic Model (CDSM). The parsers used in this model are a constituency parser (Steedman, 2001), logical parser Paradox (Claessen and Sörensson, 2003) and the DRS parser Boxer (Bos, 2008).

3.1 Features

Our model uses 25 features in total. Due to space constraints we cannot describe them all in detail here. Instead we group the features as follows:

- Lexical features: word overlap, proportional sentence length difference.
- POS: noun overlap, verb overlap.
- Logical model: instance overlap, relation overlap.
- DRS: agent overlap, patient overlap, DRS complexity.
- Entailments: binary features for: neutral, entailment and contradiction predictions.
- CDSM: The cosine distance between the element wise addition of the vectors in each sentence is used.
- Synsets (WordNet): The distance of the closest synsets of each word in both sentences, and the distance between the noun synsets.
- Named entity: overlap between named entities[1].

For a complete detailed overview we refer to the paper describing the semantic similarity system (Bjerva et al., 2014), or for even more detail (van der Goot, 2014).

3.2 Maximum Entropy Models

We will compare two different maximum entropy models. The maximum entropy implementation of Scikit-Learn (Pedregosa et al., 2011) is used.

The first maximum entropy model is a **binary** model that also outputs a probability. From this model, the normal binary output is not used, instead we use the estimated probability that something is a paraphrase. Using this value, we can set our own threshold to have more control on the final output.

The second maximum entropy model is a **multi-class** model. This classifier is based on the 6 different classes in our data, and thus outputs 6 probabilities. We use the similarity score of each class as weight to convert all probabilities to one probability. For each class we multiply the similarity score with the probability that our model predicts for this class. The results of the 6 classes are then summed to get a single probability. This classification model uses more specific training data, thus it should have a more precise output.

3.3 Normalization

A normalization approach very similar to that described by Han et al. (2013) is used to try to improve the parses. This normalization consists of three steps.

[1]This is the only feature not present in the original semantic similarity system

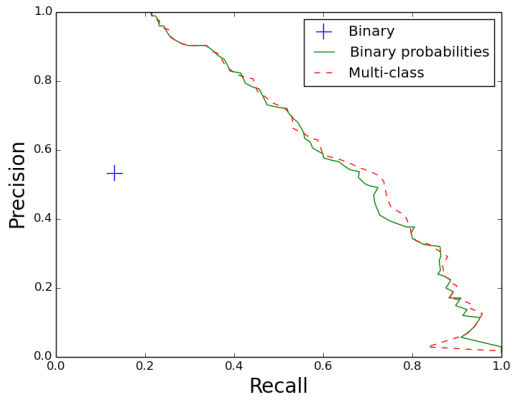

Figure 1: Precision and recall for the different classifiers.

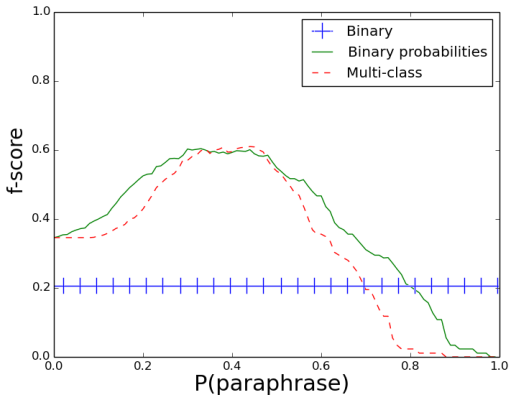

Figure 2: F-Score for the different classifiers. P is the threshold that decides if a sentence pair is a paraphrase.

The first step is to decide which tokens might need a correction, this is decided by a dictionary lookup in the Aspell dictionary[2].

The second step is the generation of possible corrections for every misspelled word. For this, the Aspell source code is adapted to lower the costs of deletion in its algorithm, because we assume words are often typed in an abbreviated form in this domain.

The last step is the ranking of the candidates. Here we use a different approach than the traditional approach. Instead of using a static formula to predict the probability of each candidate, we want to use a more flexible approach. Google N-gram probabilities (Brants and Franz, 2006), Aspell scores and dictionary lookups are combined using logistic regression. To adjusts the weights of the regression model, 200 sentences are normalized manually. The resulting model is then applied to all the other sentences.

This normalization approach does not reach a perfect accuracy, and normalizing a sentence might remove meaningful information. So instead of using the normalization as straightforward pre processing of the data, we use the raw and the normalized sentence in the model. For each feature, scores are calculated for both versions of the sentence. The highest of these scores be used as input for our maximum entropy model.

4 Evaluation

This chapter is divided in the two sub tasks of paraphrase detection and similarity prediction. A strong baseline is used, namely a state-of-the art model for clean text: a logistic regression model that uses simple lexical overlap features (Das and Smith, 2009).

4.1 Paraphrase Detection

The evaluation is done on expert annotations, which are only available for the test set. The binary and multi-class classifiers are evaluated separately. Additionally, we also tried to improve the system by using normalization.

The precision and recall of both classifiers is plotted in Figure 1. In this graph the differences are barely visible, therefore it looks like both models are approximately equal.

If we look at the F-scores of Figure 2, the differences are bigger. The highest F-scores of both classifiers are 0.604 and 0.610 for respectively the binary and the multi-class classifier. Both classifiers outperform the baseline F-score of 0.583.

These graphs also show that the default output of the binary deos not perform well, so it is really necessary to use the probabilities.

4.1.1 Feature Comparisons

We use the same grouping for features as in 3.1. The absolute weights of all features within each group are summed. For the multi-class classifier the weights are averaged over all 6 classes. Also an ablation experiment is done. An overview this evaluation is shown in Table 1.

In the ablation experiments we see that it is not always better to use more features. Especially the

[2] www.aspell.net

	Weights		Ablation	
Feat. group	**Binary**	**Multi**	**Binary**	**Multi**
Lexical	2.43	1.65	**0.601**	0.598
POS	0.79	0.71	0.600	0.600
Log. model	0.74	1.61	0.573	**0.606**
DRS	3.57	1.88	0.551	0.553
Entailments	0.51	2.79	0.584	0.589
CDSM	5.29	3.63	0.538	0.523
Synsets	0.49	0.63	0.588	0.584
NE	0.06	0.09	0.597	0.599
All	-	-	0.600	0.604

Table 1: Absolute weights of the feature groups and feature group ablation F1-Scores.

logical model should be left out in the multi-class entropy model. The models differ in some aspects, whereas some features are important for both. More specifically, we can see that the parsers outputs and lexical features are more important for the multi-class model, while the other features are more important for the binary model.

4.1.2 Normalization

After the normalization of the sentences, we run the systems again. These runs are not plotted in the graphs, because the differences are small. Despite the small differences, there is one little performance boost on the top-runs of the multi-class classifiers, resulting in the highest F-score of 0.62.

4.2 Semantic Similarity Prediction

Even though we do not have real semantic similarity training data, we simulate semantic similarity using the amount of the positive judgements per sentence pair. Our system is evolved from a semantic similarity prediction system, so this model should work well for this task. The Pearson correlation between the different annotations of experts (test) and crowdsourcing (training) is 0.735.

For this sub task we will also try different heuristics using both our classifiers. We start with the multi-class classifier, because it is trained to give back a similarity score. The model produces probabilities for each class, the class with the highest probability is used as output. We call this the `Highest P` method.

Another model can be built using the predicted weights, similar to section 3.2. We refer to this as the `Weighted` method.

	Baseline	Highest P	Binary P	Weighted
R	0.511	0.416	0.508	0.515

Table 2: Pearson correlation (R) for the different similarity prediction approaches.

Besides the multi-class classifier, we also trained a binary classifier. The only way for this classifier to output a degree score, is using the probability. This is called `Binary P`.

Only the weighted method beats the baseline. Results of all three approaches and the baseline can be found in Table 2.

5 Conclusion

The main conclusion to draw from these experiments is that by using deep semantic features, we can achieve a maximum F-score of 0.61 on the paraphrase detection task. By using normalization we can improve this F-score to 0.62.

Following from this, it is safe to conclude that a semantic similarity prediction system can be used in paraphrase detection reasonably well. Our system had an average result on this shared task (10th out of 18 teams)[3]. The advantage of this system is that it can be created easily from existing tools.

Unsurprisingly, the results on the semantic similarity task were better (6th out of 13 teams). Even though the gold data does not represent a real semantic similarity, but a scale of positive annotations of the paraphrase detection task.

The source code of our system has been made publicly available[4].

Acknowledgements

This paper is part of the 'Parsing Algorithms for Uncertain Input' project, supported by the Nuance Foundation.

We would like to thank the organizers of the shared task (Xu et al., 2015). Additionally, we would also like to thank the anonymous reviewers and Johannes Bjerva for the valuable feedback on this paper.

[3] `http://alt.qcri.org/semeval2015/task1`

[4] `https://bitbucket.org/robvanderg/sem15`

References

Johannes Bjerva, Johan Bos, Rob van der Goot, and Malvina Nissim. 2014. The Meaning Factory: Formal semantics for recognizing textual entailment and determining semantic similarity. *SemEval 2014: International Workshop on Semantic Evaluation*, pages 642–646.

Johan Bos. 2008. Wide-coverage semantic analysis with Boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286.

Thorsten Brants and Alex Franz. 2006. Web 1T5-gram corpus version 1.1.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Koen Claessen and Niklas Sörensson. 2003. New techniques that improve MACE-style finite model finding. In *Proceedings of the CADE-19 Workshop: Model Computation-Principles, Algorithms, Applications*, pages 11–27.

Dipanjan Das and Noah A Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 468–476.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *SemEval-2014*.

Brendan O'Connor, Michel Krieger, and David Ahn. 2010. Tweetmotif: Exploratory search and topic summarization for Twitter.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534.

Mark Steedman. 2001. *The Syntactic Process*.

Rob van der Goot. 2014. Automatic estimation of semantic relatedness for sentences using machine learning. Master's thesis, University of Groningen.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

Wei Xu, Chris Callison-Burch, and William B. Dolan. 2015. SemEval-2015 Task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval)*.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Kostas Tsioutsiouliklis. 2011. Linguistic redundancy in Twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 659–669.