

# CDTDS: Predicting Paraphrases in Twitter via Support Vector Regression

**Rafael Michael Karampatsis**

Department of Informatics

University of Edinburgh

10 Crichton Street, EH8 9AB Edinburgh, United Kingdom

mpatsis13@gmail.com

## Abstract

In this paper we describe a system that recognizes paraphrases in Twitter for tweets that refer to the same topic. The system participated in Task1 of SEMEVAL-2015 and uses a support vector regression machine to predict the degree of similarity. The similarity is then thresholded to create a binary prediction. The model and experimental results are discussed along with future work that could improve the method.

## 1 Introduction

Recently, Twitter has gained significant popularity among the social network services. Lots of users often use Twitter to express feelings or opinions about a variety of subjects. Analysing this kind of content can lead to useful information for fields such as personalized marketing or social profiling. However, such a task is not trivial, because the language used on Twitter is often informal, presenting new challenges to text analysis.

Task1 of SEMEVAL-2015 (Xu et al., 2015) focuses on recognition of paraphrases and semantic similarity in Twitter i.e., recognizing if two tweets are alternative linguistic expressions of the same, or similar, meaning (Bhagat and Hovy, 2013). The task is based on a crowdsourced corpus of 18000 pairs of paraphrases and non-paraphrased sentences from Twitter (Xu et al., 2014) and each pair consists of two tweets from the same topic. A label is provided with each pair, which is the number of yes votes from 5 crowdsourced annotators when asked if the second tweet is a paraphrase of the first one.

### Paraphrase Example:

Roberto Mancini gets the boot from Man City

Roberto Mancini has been sacked by Manchester City with the Blues saying

### Non-Paraphrase Example:

WORLD OF JENKS IS ON AT 11

World of Jenks is my favorite show on tv

Figure 1: Examples of both a paraphrase and a non-paraphrase pair of the data.

The method utilizes a support vector regression machine (SVR). The regression model tries to predict the degree of semantic similarity between two tweets, by assuming that it can be represented by the probability that random human annotators would annotate the pair as a paraphrase. The predicted value is transformed into a binary decision via a threshold.

Section 2 describes the data provided by the organizers. Sections 4 and 5 present the system and its performance respectively. Finally, Section 6 provides ideas for future work and Section 7 concludes.

## 2 Data

The objective of this task is to predict whether two sentences from Twitter sharing the same topic, imply the same or very similar meaning and optionally a degree score between 0 and 1. In Figure 1, a paraphrase and a non-paraphrase example taken from the task website are illustrated.

The organizers released a training (Train) and a development set (Dev), both labeled and they also provided a test set (Test) for the task. To collect

Set	Size	Paraphrase	Non-Paraphrase	Debatable
Train	13693	3996	7534	1533
Dev	724	948	2672	585
Test	972	175	663	134

Table 1: Class distribution of the train, development and test sets.

the data they used Twitter’s public API<sup>1</sup> to crawl trending topics and their associated tweets (Xu et al., 2014). Annotation of the collected tweets was performed via crowdsourcing (*Amazon Mechanical Turk*). From each topic, 11 random tweets were selected and 5 different annotators were used. One of the 11 tweets was randomly selected as the original sentence. The annotators were asked to select which of the remaining 10 tweets have the same meaning as the original one. Each topic’s pairs are annotated with the number of annotators that voted for them. Problematic annotators were removed by checking their Cohen’s Kappa agreement (Artstein and Poesio, 2008) with other annotators. Agreement with an expert annotator on 972 sentence pairs (test set) was also measured and the Pearson correlation coefficient was 0.735 although the expert annotator had actually used a different scale for the annotation. Both Train and Dev were collected from the same time period while Test was collected from a different time period.

Table 1 illustrates the class distribution of the data. The task organizers have stated that when a pair has either 1 or 0 votes it should be considered a non-paraphrase, while pairs that have 3, 4, and 5 votes should be considered as paraphrases. Pairs that have exactly 2 votes are assumed debatable and the organizers suggest that they should be discarded. We can observe that all the data sets have a very similar distribution and that the majority class is in all cases the non-paraphrase one with about 60% of the data (debatable instances included).

### 3 Previous Work

Measuring semantic text similarity has been a research subject in natural language processing, information retrieval and artificial intelligence for many years. Most works have focused on the document

<sup>1</sup><https://dev.twitter.com/docs/api/1.1/overview>

level (i.e., comparing two long texts or comparing a small text with a long one). Recently, there has been growing interest at the sentence level, specifically on computing the similarity of two sentences. The most related task to computing tweets similarity is the computation of sentence similarity.

According to (Han et al., 2012), there are three main approaches for sentence similarity. The first is based on a vector space model (Meadow, 1992) that models the text as a “bag of words” and represents it using a vector, and the similarity between the two texts is computed as the cosine similarity of their vectors. The second approach relies on the assumption that the words or expressions of two semantically equivalent sentences should be able to be aligned. The quality of this alignment can then be used as a similarity measure. When this approach is utilized, words from the two sentences are paired (Mihalcea et al., 2006) by maximizing the summation of the word similarity of the resulting pairs. Finally, the third and final approach utilizes machine learning and combines different measures and features (such as lexical, semantic and syntactic features) which are supplied to a classifier that learns a model on the training data.

The unique characteristics of Twitter present new challenges and opportunities for paraphrase research (Xu et al., 2013; Ling et al., 2013). Most of the work has focused on paraphrase generation (Xu et al., 2013; Ling et al., 2013) in order to use it for text normalization. However, the task organizers (Xu et al., 2014) created a dataset, implemented a system and reimplemented several baselines and state-of-the-art systems for sentence paraphrase recognition. They showed that their method, which combines a previous system with latent space achieves at least as good results as state-of-the-art systems.

### 4 System Overview

The main objective of the implemented system is to classify pairs of tweets from the same topic as semantically similar or not. The approach used differs from previous works because it models the problem as a regression task first and then as a classification task, while typical approaches treat the problem as a classification task (usually binary since debatable pairs are discarded). The main inspiration for this

<b>Number of Votes</b>	0	1	2	3	4	5
<b>Label Value</b>	0	0.2	0.4	0.6	0.8	1

Table 2: Mapping of the number of positive votes from the annotators to real valued labels.

approach comes from the observation that for example, pairs that got voted from 3 of the annotators will not be as similar as pairs that got voted from 5. Treating these instances in the same way is very likely to confuse the model. The regression approach is a possible way to avoid this effect since instances with different number of votes will not just use different values but will have a relation between their values. For example, instances that got 5 votes will use a better score as their label than instances that got 4 or 3.

To extract this relation from Train data, the ratio of positive votes against the total number of annotators (5) for each pair was used to create the labels. The debatable instances correspond to exactly 2 votes from the human annotators, which maps to 0.4. These instances were discarded as the organizers suggested. This resulted in the use of the values shown in Table 2 as labels.

An SVR with a linear kernel function<sup>2</sup> was used to predict the degree of similarity between the tweet pairs. For each training instance (i.e. a tweet pair) a feature vector is supplied to the regression model<sup>3</sup> along with the corresponding label. The output of the SVR can be used for classification by using a threshold. 0.35 was chosen as the classification threshold as it belongs to the debatable space and it is slightly less than 0.4, in order to increase the recall of the minority class (Paraphrases). However, the threshold could be tuned using cross validation on the training data or by testing on the development set for better results.

#### 4.1 Data Preprocessing

Preprocessing can greatly affect the performance of a system. The tweets were passed through a Twitter specific tokenizer and part-of-speech (POS) tagger (Ritter et al., 2011) by the organizers. We converted

<sup>2</sup>The LIBLINEAR distribution (Fan et al., 2008)

<sup>3</sup>The regression model uses L2-regularized regression with the default parameter C=1.

all the tweets to lower case and stopwords were removed using the NLTK (Loper and Bird, 2002) stopwords list. Moreover, we removed the tokens of the topic since they always exist in both tweets. Finally, we applied stemming to each one of the remaining tokens and the stemmed representations are stored.

#### 4.2 Feature Engineering

In this section the features used in the model will be described in detail. We made two submissions. Both share the same features except for the sentiment matching feature.

##### 4.2.1 Lexical Overlap Features

A very popular and competitive baseline is to use lexical overlap features (Das and Smith, 2009). These features use the unigrams, bigrams and trigrams of the sentences, both with and without stemming. The cardinality of the intersection of the n-grams between each pair of tweets as a proportion of the length of each tweet is used as a feature. The harmonic mean of these two values is also calculated and used as a feature. These three types of features for each n-gram size were named precision, recall and F1 (harmonic mean of precision and recall) by Das and Smith (2009).

##### 4.2.2 Ratio of the Tweets Length in Tokens

The ratio of the length of the shortest tweet in the pair divided by the length of the longer tweet is used as a feature. This feature is used because if the tweets differ a lot in length then they will probably not have similar meaning.

##### 4.2.3 Overlap of POS Tags

Similar to the lexical overlap features the overlap of POS tags of unigrams, bigrams and trigrams is checked and a total of 9 features is created. For example the tweet “Wow EJ Manuel” contains the following two POS bigrams: UH NNP and NNP NNP.

##### 4.2.4 Overlap of Named Entities Tags (NE)

Similar to the lexical overlap features the overlap of NE is checked and three features are created.

##### 4.2.5 GloVe Word Vectors Similarity

Vector space representations of words have succeeded in capturing semantic and syntactic regularities using vector arithmetic (Pennington et al., 2014;

Mikolov et al., 2013). The word vectors from GloVe (Pennington et al., 2014) were used to calculate the semantic similarity between tokens of the two sentences by measuring their cosine similarity. The word vectors utilized were created from a corpus of 2B tweets which contains 27B tokens. Experiments on the development set were also done with vectors from Wikipedia2014 + Gigawords5 (about 6B tokens) but were not used for submission since they performed worse than the Twitter ones.

The calculation of these features is based on the alignment algorithm described by (Han et al., 2013). For each of the two tweets we iterate over its tokens. For each token the similarity to all the tokens of the other tweet that exist in the model is calculated and the maximum is returned. When the algorithm finishes, the maximum, minimum and average values of the matched similarities for each tweet are returned as features. This makes a total of 6 features. An additional feature is calculated by finding the similarity of the centroids of the two tweets.

#### 4.2.6 Sentiment Matching

A Twitter sentiment classifier was used to predict the sentiment of the tweets (Karampatsis et al., 2014). The feature has a value of 1 if both tweets of the pair have the same sentiment and 0 otherwise.

## 5 Experimental Results

Each system had to submit for each test set instance its prediction (paraphrase or not) (subtask1) and optionally a degree of semantic similarity (subtask2). To evaluate system performance for subtask1 the organizers used  $F_1$  against human judgements on the predictions. While for subtask2 they used the Pearson correlation of the predicted similarity scores with the human scores. Our team was ranked 9th on both subtasks<sup>4</sup> and our systems were ranked 13th and 14th on subtask1 and 15th and 16th on subtask2. Table 3 illustrates the results and rankings of our systems and the baselines. The results indicate that the sentiment feature decreases performance and should be removed from our system.

We used the official evaluation script to assess the performance of our systems on the test set for different threshold values. The results are illustrated in

<sup>4</sup>6 teams did not participate in subtask2

Figure 2. We used thresholds from 0 to 1 with a step of 0.05 except for the space  $[0.3, 0.4]$  where we used a step of 0.01. The two systems behave similarly and the best performance (0.622) was achieved from the All-Sentiment system using a threshold of 0.36.

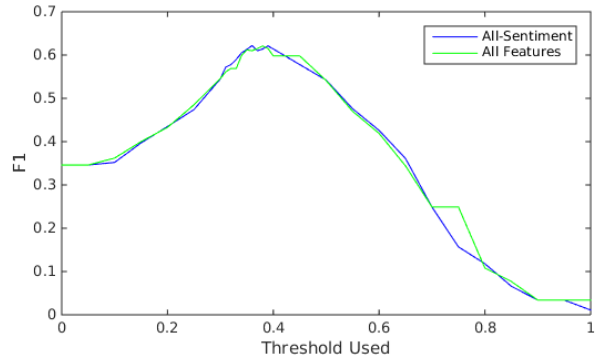


Figure 2: F1 for subtask1 on the test set for our systems using different threshold values.

## 6 Future Work

A possible direction would be to use locality sensitive hashing on the tweets (Petrovic et al., 2012) to create more features. Moreover, ordinal regression could be used to train the model (Hardin and Hilbe, 2001). The addition of a text normalization algorithm in the preprocessing step could enhance the performance of lexical overlap features and that of other methods such as wordnet, LDA (Blei et al., 2003) or LSA (Deerwester et al., 1990). Finally, the overlap of character n-grams could also be used as features.

## 7 Conclusion

We described a system that predicts semantic similarity between tweets from the same topic. The system’s aim is to identify paraphrases of a tweet on a specific topic, which is really useful in event recognition systems. It employs a support vector regression to predict the probability that human annotators would annotate a pair of tweets as a paraphrase. The predicted value is then used for binary classification by using a threshold. The system’s performance was measured on SEMEVAL-2015 Task1 and it achieves better results than the task baselines.

System	F1	F1 Rank	Precision	Recall	Pearson	Pearson Rank	maxF1	mPrecision	mRecall
All Features	0.613	13/38	0.547	0.697	0.494	15/28	0.626	0.675	0.583
All-Sentiment	0.612	14/38	0.542	0.703	0.491	16/28	0.624	0.589	0.663
LR Baseline	0.589	21/38	0.679	0.520	0.511	11/28	0.601	0.674	0.543
WTMF Baseline	0.536	28/38	0.450	0.663	0.350	26/28	0.587	0.570	0.606
Random	0.266	38/38	0.192	0.434	0.350	28/28	0.350	0.215	0.949
Human Bound	0.823	-	0.752	0.908	0.017	-	-	-	-

Table 3: Results of our systems, baselines and human annotators on the test set.

## References

- Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Comput. Linguist.*, 34(4):555–596, December.
- Rahul Bhagat and Eduard H. Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.
- Dipanjan Das and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *In Proc. of ACL-IJCNLP*.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407.
- Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.
- Bo Han, Paul Cook, and Timothy Baldwin. 2012. Automatically constructing a normalisation dictionary for microblogs. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 421–432, Jeju Island, Korea.
- Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc ebiqutycore: Semantic textual similarity systems.
- James W. Hardin and Joseph Hilbe. 2001. *Generalized Linear Models and Extensions*. College Station, Texas: Stata Press.
- Rafael Michael Karampatsis, John Pavlopoulos, and Prodromos Malakasiotis. 2014. Aueb: Two stage sentiment analysis of social network messages. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 114–118, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.
- Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2013. Paraphrasing 4 microblog normalization. In *EMNLP*, pages 73–84. ACL.
- Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. ETMTNLP ’02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Charles T. Meadow. 1992. *Text Information Retrieval Systems*. Academic Press, Inc.
- Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI06*, pages 775–780.
- Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.
- Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2012. Using paraphrases for improving first story detection in news and twitter. In *HLT-NAACL*, pages 338–346. The Association for Computational Linguistics.
- Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *EMNLP*, pages 1524–1534.
- Wei Xu, Alan Ritter, and Ralph Grishman. 2013. Gathering and generating paraphrases from twitter with application to normalization.
- Wei Xu, Alan Ritter, Chris Callison-Burch, William 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)*.