TKLBLIIR: Detecting Twitter Paraphrases with TweetingJay

Mladen Karan[1], **Goran Glavaš**[1], **Jan Šnajder**[1], **Bojana Dalbelo Bašić**[1], **Ivan Vulić**[2], **and Marie-Francine Moens**[2]

[1]University of Zagreb, Faculty of Electrical Engineering and Computing
Text Analysis and Knowledge Engineering Lab, Unska 3, 10000 Zagreb, Croatia
{goran.glavas, mladen.karan, jan.snajder, bojana.dalbelo}@fer.hr

[2]KU Leuven, Department of Computer Science
Language Intelligence & Information Retrieval Group, Celestijnenlaan 200A, Leuven, Belgium
{ivan.vulic, sien.moens}@cs.kuleuven.be

Abstract

When tweeting on a topic, Twitter users often post messages that convey the same or similar meaning. We describe *TweetingJay*, a system for detecting paraphrases and semantic similarity of tweets, with which we participated in Task 1 of SemEval 2015. TweetingJay uses a supervised model that combines semantic overlap and word alignment features, previously shown to be effective for detecting semantic textual similarity. TweetingJay reaches 65.9% F1-score and ranked fourth among the 18 participating systems. We additionally provide an analysis of the dataset and point to some peculiarities of the evaluation setup.

1 Introduction

Recognizing tweets that convey the same meaning (paraphrases) or similar meaning is useful in applications such as event detection (Petrović et al., 2012), tweet summarization (Yang et al., 2011), and tweet retrieval (Naveed et al., 2011). Paraphrase detection in tweets is a more challenging task than paraphrase detection in other domains such as news (Xu et al., 2013). Besides brevity (max. 140 characters), tweets exhibit all the irregularities typical of social media text (Baldwin et al., 2013), such as informality, ungrammaticality, disfluency, and excessive use of jargon.

In this paper we present the *TweetingJay* system for detecting paraphrases in tweets, with which we participated in Task 1 of SemEval 2015 evaluation exercise (Xu et al., 2015). Our system builds on findings from a large body of work on semantic textual similarity (STS) (Šarić et al., 2012; Sultan et al., 2014) and recent breakthroughs in distributed word representations (Mikolov et al., 2013a). We design a set of measures that capture the semantic similarity of tweets and train a support vector machine (SVM) using these measures as features. Positioning of our system at rank four among 18 teams, with only point and a half lower performance compared to the the best-performing system, suggests that STS measures are useful for detecting paraphrases in Twitter. We make our system freely available.[1]

Besides providing the description of the TweetingJay system, in this paper we analyze the evaluation setup, with special focus on the provided dataset and its subsets (train, validation, and test), and discuss the stability of the evaluation results.

2 Related Work

There is a large body of work on automated paraphrase detection; see (Madnani and Dorr, 2010) for a comprehensive overview. The majority of research efforts focus on detecting paraphrases in standard texts such as news (Das and Smith, 2009; Madnani et al., 2012) or artificially generated text (Madnani et al., 2012). State-of-the-art approaches typically combine several measures of semantic similarity between text fragments. For instance, Madnani et al. (2012) achieve state-of-the-art performance by combining eight different machine translation metrics in a supervised fashion.

A task closely related to paraphrase detection is semantic textual similarity (STS), introduced at SemEval 2012 (Agirre et al., 2012). There is now a

[1]http://takelab.fer.hr/tweetingjay

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 70–74,
Denver, Colorado, June 4-5, 2015.

significant amount of work on this task. The best performing STS systems employ various methods for aligning semantically corresponding words or otherwise quantifying the amount of semantically congruent content between two sentences (Sultan et al., 2014; Šarić et al., 2012).

In contrast, STS research on Twitter data has been scarce. Zanzotto et al. (2011) detect content redundancy between tweets, where redundant means paraphrased or entailed content. They achieve reasonable performance with SVM using vector-comparison and syntactic tree kernels. Xu et al. (2014) propose MULTIP, a latent variable model for joint inference of correspondence of words and sentences. An unsupervised model based on representing sentences in latent space is presented by Guo and Diab (2012).

3 TweetingJay

TweetingJay is essentially a supervised machine learning model, which employs a number of semantic similarity features (18 features in total). Because the number of features is relatively small, we use SVM with a non-linear (RBF) kernel. Our features can be divided into (1) semantic overlap features, most of which are adaptations of STS features proposed by Šarić et al. (2012), and (2) word alignment features, based on (a) the output of the word alignment model by Sultan et al. (2014) and (b) a re-implementation of the MULTIP model by Xu et al. (2014).

In the dataset provided by the organizers, each tweet is associated with a topic, with 10 to 100 tweet pairs per topic. An important preprocessing step is to remove tokens that can be found in the name of a topic. For example, for the topic *"Roberto Mancini"*, we trim the tweets *"Roberto Mancini gets the boot from the Man City"* and *"City sacked Mancini"* to *"gets the boot from the Man City"* and *"City sacked"*, respectively, and then compute the features on the trimmed tweets. The rationale is that, given a topic, there is an overlap in topic words between both paraphrase and non-paraphrase tweet pairs, which diminishes the discriminative power of the model's comparison features.

3.1 Semantic Overlap Features

Semantic overlap features compare the content words (nouns, verbs, adjectives, adverbs, and numbers).

Ngram overlap. We compute the number of matching n-grams between two tweets. This number is normalized by the length of the first and the second tweet, respectively, and the harmonic mean of these two measures is taken as the similarity score. These features are computed separately for unigrams and bigrams. We also compute a weighted version by weighting the matched words w with their information content:

$$ic(w) = -\log \frac{freq(w) + 1}{\sum_{w' \in C} freq(w') + 1}$$

where C is the set of all words in the corpus and $freq(w)$ is the word's frequency. We obtained the frequencies from the Google Books Ngrams (GBN) (Michel et al., 2011). In the weighted version of the ngram overlap, the overlap is normalized by the sum of information contents of all words in the first and second tweet, respectively, and the resulting similarity score is the harmonic mean of these two scores.

Greedy word alignment overlap (GWAO). To compute this feature, we iteratively pair the words – one word from each tweet – according to their semantic similarity. In each iteration we greedily select the pair of words with the largest semantic similarity, and remove the words from their corresponding tweets, until no words are left in shorter of the two tweets. The similarity between words is computed as the cosine between their corresponding 300-dimension embedding vectors obtained using `word2vec` tool (Mikolov et al., 2013b) on a 100 billion words portion of the Google News dataset. Let $P(t_1, t_2)$ be the set of word pairs obtained through the alignment on a pair of tweets (t_1, t_2) and let $vec(w)$ be the embedding vector of the word w. The GWAO score is computed as:

$$gwao(t_1, t_2) = \sum_{\substack{(w_1, w_2) \\ \in P(t_1, t_2)}} \alpha \cdot \cos\left(vec(w_1), vec(w_2)\right)$$

where α is the larger of the information contents of the two words, $\alpha = \max\left(ic(w_1), ic(w_1)\right)$. The $gwao(t_1, t_2)$ score is normalized with the sum of information contents of words from t_1 and t_2, respectively, and the harmonic mean of the two normalized scores is taken as the feature value.

Tweet embedding similarity. Linear combinations of word embedding vectors have been shown to correspond well to the semantic composition of the individual words (Mikolov et al., 2013a; Mikolov et al., 2013b). Building on this finding, we embed a tweet as a weighted sum of the embeddings of its content words, where we use information content of words as their weights:

$$vec(t) = \sum_{w \in t} ic(w) \cdot vec(w).$$

As the tweet embedding similarity, we simply compute the cosine between the corresponding tweet embeddings, i.e., $\cos\left(vec(t_1), vec(t_2)\right)$.

Topic-specific information content. While information content computed on a general corpus such as GBN indicates how informative the word is in general, we also wanted to have a measure of how informative each word is within a tweet's topic. To this end we also compute topic-specific versions of all the above features using topic-specific instead of GBN information contents.

3.2 Word Alignment Features

We adopt the word alignment features from two alignment-based systems: (1) the DLS@CU system of Sultan et al. (2014), which achieved the best performance on the STS task at SemEval 2014 (Agirre et al., 2014), and (2) our implementation of the MULTIP latent variables model (Xu et al., 2014), which utilizes the concept of an *anchor*: a pair of semantically aligned words from a paraphrased pair of tweets.

Aligned word pairs (AWP). A state-of-the art monolingual word alignment model by Sultan et al. (2014) outputs pairs of semantically aligned words between two given sentences.[2] We used the output of the DLS@CU model to generate two features: (1) the raw count of the aligned word pairs, and (2) the normalized count, which is the harmonic mean of the scores obtained by normalizing the raw count with the length of the first and second tweet, respectively. We computed two versions for both of these features, one considering all the tokens in tweets, and the other taking into account only content words.

Anchor count (ANC). We re-implemented the MULTIP model of Xu et al. (2014).[3] As anchor candidates we consider all pairs of content words from the two tweets. We use a minimalistic set of features including (1) Levenshtein distance between candidate words, (2) several binary features indicating relatedness of words (e.g., lowercased tokens match, POS-tags match), and (3) semantic similarity obtained as the cosine of word embeddings, obtained with the GloVe model (Pennington et al., 2014) trained on Twitter data.[4] To account for feature interactions, following (Xu et al., 2014), we also use conjunction features. We use the number of anchors identified by this method for a pair of tweets as a feature for our SVM model.

4 Evaluation

Each team was allowed to submit two runs on the test set provided by the task organizers (Xu et al., 2015). Participants were provided with a training set (13,063 pairs) and a development set (4,727 pairs). We used the train and development set to optimize the hyperparameters C and γ of our SVM model with the RBF kernel. For the final evaluation, the organizers used a test set of 972 tweet pairs.

Feature sets. We divided the features in three groups: (1) semantic overlap features (SO) from Section 3.1, (2) aligned word pairs (AWP) features, and (3) the anchor count feature (ANC) from Section 3.2.

Model optimization. There are three ways how the optimization of the SVM model (hyperparameters C and γ) could have been carried out: (1) training and optimization on the train set using 10-folded cross-validation, with no use of the development set (model M1); (2) training on the train set and optimization on the development set (model M2), and (3) training on the union of the train and development set using 10-folded cross-validation (model M3). Following the advice of the task organizers, we removed debatable cases from both the train and dev sets. We submitted models M1 and M2 for the official evaluation (our team name was TKLBLIIR).

[2] https://github.com/ma-sultan/monolingual-word-aligner

[3] We obtain lower results on the test set (61.3% F_1 vs. 69.6%). This is likely caused by the use of slightly different features and perhaps by differences in implementation.

[4] http://nlp.stanford.edu/projects/glove/

Team	P	R	F_1	Rank
ASOBEK	68.0	66.9	**67.4**	1
MITRE	80.6	56.9	66.7	2
ECNU	76.7	58.3	66.2	3
FBK-HLT	68.5	63.4	65.9	4
TKLBLIIR M1	64.5	67.4	**65.9**	5
TKLBLIIR M2	46.1	81.7	59.0	19
MULTIP	71.9	67.4	**69.6**	–
Baseline (log.reg.)	67.9	52.0	58.9	21
Baseline (WTMF)	45.0	66.3	53.6	28

Table 1: Official SemEval Task 1 evaluation.

	M1		M2		M3	
Features	dev	test	dev	test	dev	test
SO	63.3	63.4	**64.9**	59.0	63.3	61.5
SO+AWP	64.0	61.6	64.7	60.4	64.0	61.6
SO+ANC	60.8	**65.9**	64.6	60.8	64.5	62.5
SO+AWP+ANC	64.1	63.2	**64.9**	59.0	64.4	61.2

Table 2: Model optimization using different datasets.

4.1 Official Results

A subset of the official ranking is shown in Table 4.1. Our model M1 ranked fourth (sharing that place with FBK-HLT) in the official evaluation with a 1.5% lower F_1 score than the best-performing system. Our model M2 outperforms both baselines. The state-of-the-art model MULTIP outperforms all participating systems. There is a notable performance gap between our two runs. We believe this comes from the high sensitivity of the performance on the test set to small changes in hyperparameter values. We elaborate more on this in the next section.

4.2 Dataset Analysis

In Table 4.2 we show the performance of the models M1, M2, and M3 on the development and test set. We observe an unusual behavior for all three models: a model that performs good on the development set typically performs bad on the test set, and vice versa. Furthermore, optimal cross-validated F_1 performance on the train set is 72%, which is 7 points above the best performance on the validation set. We believe this may be indicative of significant differences in the distributions underlying the datasets.

To investigate this further, we applied the Kolmogorov-Smirnov two-sample goodness-of-fit test (K-S test) (Daniel, 1990) for each of the used features to determine whether the train set is drawn from the same distribution as the development and test set. The K-S test is a nonparametric test that determines whether two independent samples differ in some respect, both in the measure of locations (means, median) and the shapes of the distributions (skewness, dispersion, kurtosis). The assumptions for the K-S test (independence of random samples and continuous variables) are met for all our features. We tested all features at the level of significance of 0.05 and rejected the null hypothesis for all features but one (bigram overlap). This confirms our initial assumption that the features in the train set are not identically distributed to those in the test set, bringing into question the representativeness of the test set. Reasons for this may include different annotation sources (crowdsourcing vs experts) and differences in time periods of tweets. Moreover, due to differences in the datasets, the performance is very much affected by the choice of the model optimization setup.

4.3 Feature Analysis

Due to volatile performance, it is difficult to say much about which features are most useful. However, we have observed consistent performance boosts in all settings when introducing topic-specific versions of features.

5 Conclusion

We described TweetingJay, a supervised model for detecting Twitter paraphrases with which we participated in Task 1 of SemEval 2015. TweetingJay relies on features capturing semantic similarity and word alignments between tweets and achieves performance comparable to best-performing models on the task.

On the methodological side, we investigated the cause for unusual behavior of our models on the different datasets. Our preliminary statistical analysis of the datasets seems to suggest that the underlying distributions datasets are significantly different. We believe this makes the performance estimates less reliable and suggest that the results should be taken with caution.

References

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of SemEval 2012*, pages 385–393.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. pages 81–91.

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources. In *Proceedings of IJCNLP 2013*, pages 356–364.

Wayne W. Daniel. 1990. *Applied nonparametric statistics*. The Duxbury advanced series in statistics and decision sciences.

Dipanjan Das and Noah A Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL 2009*, pages 468–476.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of ACL 2012*, pages 864–872.

Nitin Madnani and Bonnie J Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of NAACL 2012*, pages 182–190.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. 2011. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS 2013*, pages 3111–3119.

Nasir Naveed, Thomas Gottron, Jérôme Kunegis, and Arifah Che Alhadi. 2011. Searching microblogs: coping with sparsity and document quality. In *Proceedings of CIKM 2011*, pages 183–188.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP 2014*, pages 1532–1541.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2012. Using paraphrases for improving first story detection in news and twitter. In *Proceedings of NAACL 2012*, pages 338–346.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: systems for measuring semantic text similarity. In *Proceedings of SemEval 2012*, pages 441–448.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. DLS@CU: Sentence similarity from word alignment. In *Proceedings of SemEval 2014*, pages 241–245.

Wei Xu, Alan Ritter, and Ralph Grishman. 2013. Gathering and generating paraphrases from twitter with application to normalization. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 121–128.

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 (TACL)*, 2(1).

Wei Xu, Chris Callison-Burch, and William B. Dolan. 2015. SemEval-2015 Task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of SemEval 2015*.

Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. 2011. Social context summarization. In *Proceedings of ACM SIGIR 2011*, pages 255–264.

Fabio Massimo Zanzotto, Marco Pennacchiotti, and Kostas Tsioutsiouliklis. 2011. Linguistic redundancy in twitter. In *Proceedings of EMNLP 2011*, pages 659–669.