Ebiquity: Paraphrase and Semantic Similarity in Twitter using Skipgram

Taneeya Satyapanich, Hang Gao and Tim Finin
University of Maryland, Baltimore County
Baltimore, MD, 21250, USA
taneeya1@umbc.edu, hanggao1@umbc.edu, finin@umbc.edu

Abstract

We describe the system we developed to participate in *SemEval 2015 Task 1, Paraphrase and Semantic Similarity in Twitter*. We create similarity vectors from two-skip trigrams of preprocessed tweets and measure their semantic similarity using our UMBC-STS system. We submit two runs. The best result is ranked eleventh out of eighteen teams with F1 score of 0.599.

1. Introduction

In this task (Wei, et al., 2015), participants were given pairs of text sequences from Twitter trends and produced a binary judgment for each stating whether or not they are paraphrases (e.g., semantically the same) and optionally a graded score (0.0 to 1.0) measuring their degree of semantic equivalence. For example, for the trending topic "*A Walk to Remember*" (a film released in 2002), the pair *A Walk to Remember is the definition of true love*" and "*A Walk to Remember is on and Im in town and Im upset*" might be judged as not paraphrases with score 0.2 whereas the pair "*A Walk to Remember is the definition of true love*" and "*A Walk to Remember is the cutest thing*" could be judged as paraphrases with a score of 0.6.

Many methods have been proposed to solve the paraphrase detection problem. Early approaches were often based on lexical matching techniques, e.g., word n-gram overlap (Barzilay and Lee, 2003) or predicate argument tuple matching (Qiu, et al., 2006). Some other approaches that go beyond simple lexical matching have also been developed. For example, (Mihalcea, et al., 2006) estimated semantic similarity of sentence pairs with word-to-word similarity measures and a word specificity measure. (Zhang and Patrick, 2005) uses text canonicalization to transfer texts of similar meaning into the same surface text with a higher probability than those with different meaning.

Many of these approaches adopt distributional semantic models, but limited to a word level. To extend distributional semantic models beyond words, several researchers have learned phrase or sentence representation by composing the representation of individual words (Mitchell and Lapata, 2010; Baroni and Zamparelli, 2010). An alternative approach by (Socher et al., 2011) represents phrases and sentences with fixed matrices consisting of pooled word and phrase pairwise similarities. (Le and Mikolov, 2014) learns representation of sentences directly by predicting context without composition of words.

In our work, we judge that two sentences are paraphrases if they have high degree of semantic similarity. We use the UMBC-Semantic Textual Similarity system (Lushan Han et al., 2013), which provides high accurate semantic similarity measurement. The remainder of this paper is organized as follows. Section 2 describes the task and the details of our method. Section 3 presents our re-

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 51–55,
Denver, Colorado, June 4-5, 2015.

sults and a brief discussion. The last section offers conclusions.

2. Our Method

To decide whether two tweets are paraphrases or not, we use a measurement based on semantic similarity values. If two tweets are semantically similar, they are judged as paraphrases, otherwise they are not. We described steps of our method as follows.

1.1. Preprocessing

Generally, tweets are informal text sequences that include abbreviations, neologisms, emoticons and slang terms as well genre-specific elements such as hashtags, URLs and @mentions of other Twitter accounts. This is due to both the informal nature of the medium and the requirement to limit content to at most 140 characters. Thus, before measuring the semantic similarity, we replace abbreviation and slang to the readable version. We collected about 685 popular abbreviations and slang terms from several Web resources[1] and combined these with the provided twitter normalization lexicon developed by Han Bo and Timothy Baldwin (2011).

After replacing abbreviations and slang terms, we remove all stop words to get our final desired processed tweets. Then we produce a set of two-skip trigrams for each tweet and name these sets as *trigram sets*. We adapted the skip-gram technique from (Guthrie, et al., 2006).

Take the tweet "*Google Now for iOS simply beautiful*" as an example, after removing stop word s, we get *'Google Now iOS simply beautiful'*. Then a two-skip trigram set is produced: {*'Google Now iOS', 'Now iOS simply', 'iOS simply beautiful', 'Google iOS simply', 'Google simply beautiful', 'Now simply beautiful', 'Google Now beautiful', 'Google Now simply', 'Now iOS beautiful'}*, which is referred as trigram set. We transform every raw tweet into its processed version and then corresponding trigram set.

[1] These included http://webopedia.com, http://blog.-mltcreative.com and http://internetslang.com and others.

1.2. LSA Word Similarity Model

Our LSA word similarity model is a revised version of the one we used in the 2013 and 2014 SemEval semantic text similarity tasks (Han, et al., 2013, Kashyap et al., 2014). LSA relies on the fact that semantically similar words (e.g., cat and feline or nurse and doctor) are more likely to occur near one another in text. Thus evidence for word similarity can be computed from a statistical analysis of a large text corpus. We extract raw word co-occurrence statistics from a portion of a 2007 Stanford WebBase dataset (Stanford, 2001).

We performed part of speech tagging and lemmatization on the corpus using the Stanford POS tagger (Toutanova et al., 2000). Word/term co-occurrences were counted with a sliding window of fixed size over the entire corpus. We generate two co-occurrence models using window sizes ±1 and ±4. The smaller window provides more precise context which is better for comparing words of the same part of speech while the larger one is more suitable for computing the semantic similarity between words of different syntactic categories.

Our word co-occurrence models are based on a predefined vocabulary of 22,000 common English open-class words and noun phrases, extended with about 2,000 verb phrases from WordNet. The final dimensions of our word/phrase co-occurrence matrices are 29,000×29,000 when words/phrases are POS tagged. We apply singular value decomposition on the word/phrase co-occurrence matrices (Burgess 1998) after transforming the raw word/phrase co-occurrence counts into their log frequencies, and select the 300 largest singular values. The LSA similarity between two words/phrases is then defined as the cosine similarity of their corresponding LSA vectors generated by the SVD transformation.

To compute the semantic similarity of two text sequences, we use the simple *align-and-penalize* algorithm described in (Han et al., 2013) with a few improvements. These improvements include some sets of common disjoint concepts and an enhanced stop word list.

1.3. Features

For two trigram sets, we compute the semantic similarity of every possible pair of trigrams in these two sets using the UMBC Semantic Textual

Similarity system. For each pair of tweet (T1 and T2), six features are produced as:

- Feature1 = semantic similarity value between each pair of tweets (whole sentence with abbreviation and slangs replaced, and stop words removed)
- Feature2 = $Max(Max(sim(T1,T2)))$
- Feature3 = $Max(Max(sim(T2,T1)))$
- Feature4 = $Avg(Max(sim(T1,T2)))$
- Feature5 = $Avg(Max(sim(T2,T1)))$
- Feature6 = the weighted average on length of tweets of two averages above.

1.5. Training

We used the LIBSVM system (Chang and Lin, 2011) for training a *logistic regression* model and a *support vector regression* model. We run a grid search to find the best parameters for both models. All training data (13,063 pairs of tweets) were used to train the models without discarding any debatable data. We tested the contribution for of each of the features through ablation experiments on the development data in which each feature was deleted in each experimental run. Table 1 shows the statistical results for each feature ablation run.

Feature deleted	F1	Precision	Recall
Feature 1	0.7	0.709	0.728
Feature 2	0.697	0.706	0.726
Feature 3	0.697	0.706	0.726
Feature 4	0.691	0.700	0.722
Feature 5	0.696	0.706	0.726
Feature 6	0.695	0.705	0.725

Table 1. Performance of our system on runs against the development data in which each feature was removed.

From Table 1, we can see that the feature of lowest performance is Feature 1, the semantic similarity computed with entire tweets without using the skip-gram technique. But we still keep Feature 1 since performance of these six features is not significantly different. We show the performance of each model on development data in Table 2.

Model	F1	Precision	Recall
Logistic Regression	0.697	0.706	0.726
Support Vector Regression	0.691	0.707	0.726

Table 2. Performance of system on development data.

Since the performance of both systems is almost the same, we decide to submit one run of each system.

3. Results and Discussions

We submit two runs: Run_1 (Logistic Regression) obtained an F1 score of 0.599, precision score of 0.651 and recall score of 0.554, and Run_2 (Support Vector Regression), which received an F1 of 0.590, precision of 0.646, and recall of 0.543. When ranked, we are in the eighteenth (Run_1) and the nineteenth (Run_2) out of the 38 runs. The first rank has F1 score of 0.674. The full distribution of F1 score is shown in Figure 1. The relatively low ranking of our system might be the result of several factors.

First factor is the prevalence of neologisms, misspellings, informal slang and abbreviations in tweets. Better preprocessing to make the tweets closer to normal text might improve our results.

Another factor is the UMBC STS system. Examples of input on which UMBC STS system perform poorly are shown in Table 3. We can group these into two sets, each associated with problem in performing the paraphrase task.

The first problem is that a slang word may have different meanings when it is used in different genres. As we can see in the first example in Table 3, *'bombs'* does not mean '*a container filled with explosive'* but is a synonym of *'home runs'* when mentioned in a sports or baseball context. We can recognize this meaning by reading sport articles but it is not included in any dictionaries or WordNet. Thus our system predicts that the two tweets, each containing either 'bombs' or 'home runs', have low semantic similarity and thus are not paraphrases.

The second problem involves out-of-vocabulary words, such as the named entities found in the examples in Table 3. Tweet 2 of the second example

'NOW YOU SEE ME and AFTER EARTH Cant Outpace FAST FURIOUS 6' is full of movie names whose meanings our STS system cannot recognize. We can solve this problem by adding name entity recognition to the system. Another potential solution would be to adopt a simple string-matching component. With string matching, we may handle those out-of-vocabulary words situations similar to the third and fourth example. We can match *'orr'* and *'chara'* between two tweets of the third example and *'new ciroc'* in the fourth example.

To improve our STS performance, which is trained on a corpus that mostly consisted of reasonably well-written narrative text, we need to expand training corpus. Training a LSA model on a collection of tweets or a mixture of tweets and narrative text, and adding name entity recognition process may lead to better results.

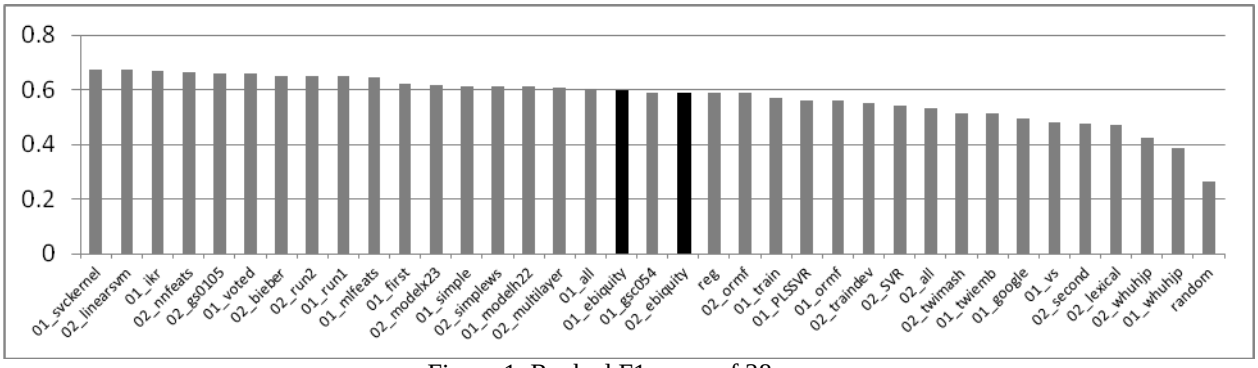

Figure 1. Ranked F1 score of 38 runs

#	Tweet 1	Tweet 2	System	Gold
1	chris davis is 44 with two bombs	Chris Davis has 2 home runs tonight	False	True
2	I wanna see the movie after earth	NOW YOU SEE ME and AFTER EARTH Cant Outpace FAST FURIOUS 6	True	False
3	Orr with a big hit on Chara	I keep waiting for the chara vs orr fight	False	True
4	New Ciroc Amaretto I NEED THAT	Oh shit I gotta try that new ciroc flavor	False	True

Table 3. Examples of input pairs on which our system performed poorly

4. Conclusion

We describe our system submitted in participating the *SemEval 2015 Task 1 Paraphrase and Semantic Similarity in Twitter*. We preprocess tweets using two-skip trigrams to produce sets of possible trigrams and measure their semantic similarity using the UMBC-STS system. We computed the statistical value as maximum and average of each pair and use two regression models; logistic regression and support vector regression. Our best performing run achieved an F1 score of 0.599 and was ranked eleventh out of eighteen teams.

Acknowledgments

Partial support for this research was provided by grants from the National Science Foundation (1228198 and 1250627) and a grant from the Maryland Industrial Partnerships program.

References

Marco Baroni and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010).

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies(HLT-NAACL)

William Blacoe. and Mirella Lapata 2012. A comparison of vector-based representations for semantic composition, Proceedings of EMNLP, Jeju Island, Korea, pp. 546-556.

Han, Bo, and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a# twitter. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.

Curt Burgessa, Kay Livesayb and Kevin Lundb 1998. Explorations in context space: Words, sentences, discourse. Discourse Processes, 25:211–257.

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, 2011.

David Guthrie, Ben Allison, Wei Liu, Louise Guthrie, Yorick Wilks. 2006. "A closer look at skip-gram modelling." In Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006), pp. 1-4. 2006.

Lushan Han, Tim Finin, Paul McNamee, Anupam Joshi and Yelena Yesha, Improving Word Similarity by Augmenting PMI with Estimates of Word Polysemy, IEEE Transactions on Knowledge and Data Engineering, IEEE Computer Society, v25n6, pp. 1307-1322, 2013.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Johnathan Weese. 2013. UMBC EBIQUITY-CORE: Semantic Textual Similarity Systems, In Second Joint Conf. on Lexical and Computational Semantics. Association for Computational Linguistics , June.

Lushan Han, Schema Free Querying of Semantic Data, Ph.D. Dissertation, University of Maryland, Baltimore County, August 2014.

Abhay Kashyap, Lushan Han, Roberto Yus, Jennifer Sleeman, Taneeya Satyapanich, Sunil Gandhi and Tim Finin. 2014. Meerkat Mafia: Multilingual and Cross-Level Semantic Textual Similarity Systems, Int. Workshop on Semantic Evaluation, Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity, Proceedings of the National Conference on Artificial Intelligence (AAAI 2006), Boston, Massachusetts, pp. 775-780

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. Cognitive Science, 34(8).

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2006. Paraphrase recognition via dissimilarity significance classification. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, pages 18–26, Sydney, Australia, July. Association for Computational Linguistics.

Le, Quoc V., and Tomas Mikolov. "Distributed representations of sentences and documents." arXiv preprint arXiv:1405.4053 (2014).

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Advances in Neural Information Processing Systems (NIPS 2011).

Stanford. 2001. Stanford WebBase project. http://bit.ly/WebBase.

Kristina Toutanova, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, and Michel Galley. 2000. Stanford log-linear part-of-speech tagger. http://nlp.stanford.edu/software/tagger.shtml.

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

Yitao Zhang and Jon Patrick. 2005. Paraphrase identification by text canonicalization. In Proceedings of the Australasian Language Technology Workshop 2005, pages 160–166, Sydney, Australia, December.