Using Paraphrases to Improve Tweet Classification: Comparing WordNet and Word Embedding Approaches

Quanzhi Li, Sameena Shah
R&D, Thomson Reuters
3 Times Square, NYC, NY 10036
{quanzhi.li, sameena.shah}
@thomsonreuters.com

Mohammad Ghassemi
EECS, MIT
Cambridge, MA 02139
ghassemi@mit.edu

Rui Fang, Armineh Nourbakhsh, Xiaomo Liu
R&D, Thomson Reuters
3 Times Square, NYC, NY 10036
{rui.fang, armineh.nourbakhsh, xiaomo.liu}
@thomsonreuters.com

***Abstract*—two of the major problems in social media message classification are the data sparseness issue and the high degree of lexical variation. Paraphrases, or synonyms, are alternative ways of expressing the same meaning using different lexical variations. In this study, we try to use paraphrases to improve tweet topic classification performance. We explored two approaches to generating paraphrases, WordNet, which is a lexical database grouping English words into sets of synonyms, and word embeddings, which are learned from millions of tweets and billions of words. Our experiment shows that using paraphrases can improve the topic classification task, and the word embedding approach outperforms the WordNet method. To our knowledge, this is the first study exploiting paraphrases for tweet classification.**

Keywords-paraphrases; synonyms; vocabulary mismatch; word embedding; WordNet; tweet topic classification

I. INTRODUCTION

Tweets are about a quite variety of topics, and users or applications are usually only interested certain topic categories, such as sports, politics or business. Therefore, identifying exactly which tweet is about which topic category is necessary. Classifying tweets poses new challenges, because tweets are short, noisy, and less topic-focused. Many classification tasks on short text, such as tweet, fail to achieve high accuracy due to data sparseness issue. Several studies have attempted to overcome this problem by exploiting external knowledge, such as using search engines to expand the tweet context, and using the embedded link to get the content of that page to enrich the tweet data. Both approaches are time-consuming and are not suitable for real-time applications. Another challenge for text categorization is the vocabulary mismatch problem. The words "pleasure" and "happiness" will not match unless a synonym/paraphrase list is predefined. This problem is more serious for short text, such as tweet, since they have very limited context to use.

One possible solution for addressing the vocabulary mismatch problem well and the data sparseness issue to certain degree is using external sources of structured semantic knowledge. This is one of the two approaches we explored in this study, which uses WordNet [7, 12] as the external knowledge source. The problem with a semantic knowledge base is that it may not be suitable for social media messages. Tweets have many terms and abbreviations that are not in the knowledge bases, and these knowledge bases are usually not updated often enough to reflect the dynamics of social media. This is why we also explored the second approach, using word embeddings learned directly from tweets. Compared to a structured knowledge base, the word embedding approach can reflect the word relationship in social media more dynamically, and cover more terms and their variations. For a given a term, we can find its semantically closest terms by calculating the cosine similarity between their embedding vectors. Those terms are then used as this term's paraphrases in the tweet topic classification task. We compared these two approaches in this study, to see if they improve the tweet classification performance, and which method performs better.

In the following sections, we first describe WordNet and word embeddings, then we present our evaluation method and experiment result, followed by related studies

II. GENERATING PARAPHRASES USING WORDNET AND WORD EMBEDDING

A. Paraphrase from WordNet

WordNet is a lexical database for the English language. It groups words into sets of synonyms called synsets, provides short definitions, and records a number of relations among these synonym sets. WordNet version 3 contains around 118,000 synsets and is used in this study to find paraphrases. There are four types of words in WordNet, nouns, verbs, adjectives and adverbs. In this study, we just use nouns and verbs, since adjectives and adverbs are not good at differentiating text into different topic categories. There are also other paraphrases sources, such as Microsoft Research paraphrase tables, which are a set of paraphrase pairs automatically extracted from news articles, and paraphrases from Callison-Burch [2], which are syntactically constrained. We chose WordNet because it covers more concepts and it has been used by many previous studies.

Because we only use nouns and verbs of WordNet synsets, the TweetNLP package [11] is used to identify the part-of-speech tag for each word in a tweet. For nouns and verbs, we identify their paraphrases from the synsets. For words with multiple meanings, the most common sense is chosen and the corresponding synset is used to generate paraphrases. At prediction time, when calculating the similarity between a tweet and the topic categories, if there

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

is no match for an original term (a noun or verb), its paraphrases are checked, and the term weight of the original term is used for its paraphrases.

B. Paraphrases from Word Embeddings

Word embedding is a low-dimensional, dense and real-valued vector for a word. They are usually generated from a large corpus, and the embeddings of a word capture both the syntactic structure and semantics of the word. Traditional bag-of-words and bag-of-n-grams hardly capture the semantics of words, or the distances between words. One implementation of the word embedding model is the word2vec model from Mikolov et al. [5, 6], which is used in this study. Generating word embeddings from text corpus is an unsupervised process. To get high quality embedding vectors, a large amount of training data is necessary. After training, each word, including all hashtags in the case of tweet, is represented by a low-dimensional, real-valued vector.

Different from the WordNet approach, which uses paraphrases for just nouns and verbs, in this approach, we generate paraphrases for nouns, verbs, proper nouns (named entities), and hashtags. We can generate paraphrases for hashtags and proper nouns because the word embedding model contains embeddings for them. In contrast, WordNet does not have hashtags or proper nouns in their synsets.

A term's paraphrases were generated as follows: the word's 300-dimensional word embeddings were obtained by querying the trained embedding model; the cosine similarity score was calculated between this word's embedding vector and the embedding vector of each term in this model. To make the paraphrases amount manageable, for each term, at most 15 paraphrases were generated. The candidates with the highest scores were selected, and their similarity scores should also be greater than 0.6. These thresholds were obtained empirically.

Similarly to the WordNet approach, at prediction time, if there is no match for an original term, its paraphrases are used. But here we also consider hashtags and proper nouns, in addition to nouns and verbs.

C. Evaluation

The task in this study is to classify a tweet into one of the ten topic categories. The topic categories and the evaluation baseline are described below.

Topic categories: We used a predefined list of ten topics in this study. These topics are defined by journalists and they reflect the requirements of users from a real-time tweet topic classification system. The ten categories are: Business/Finance, Science/Technology, Politics, Entertainment, Health/Medical, Law/crime, Crisis (War/Disaster), Weather, Sports, and Society/Life. Although the experiments were conducted on this specific list of topics, we believe the proposed approaches may be applied, directly or with some adjustment, to other applications with different topics.

Baseline method: Many learning algorithms, such as SVM, KNN, Rocchio classifier and logistic regression, have been used in many text classification problems, and achieved good results. In this study, the baseline classifier is the Rocchio classifier, which is basically a K-mean algorithm. This model uses the text presented in the training tweets to build the classifier. It has the following steps:

- The tweet text is preprocessed as follow: dates are converted to a symbol; all ratios are replaced by a special symbol; numbers are normalized to a symbol; all URLs and mentions are removed; all special characters except hashtags are removed; stop words are removed.
- For each term in a class (topic category), a tf.idf value is calculated as its weight. For calculating tf.idf, each class is treated as a document.
- A centriod document (tweet) is generated based on all terms of a class to represent this category.

At prediction time, for a test tweet, a cosine similarity value is calculated against each topic category. The category with the highest score is the class the tweet should belong to.

Evaluation metrics: We use the classic metrics for classification task for evaluation: precision, recall, and F (F1) measure.

III. EXPERIMENTS

A. Data Set for Generating Word Embeddings

Table 1 shows the statistics of the tweets used for training the word embedding model. Only English tweets are included in this study. Totally, there are about 200 million tweets and 2.9 billion words that are used to train the word embedding model. Finally, we got about 2 million unique words.

Table 1. Basic Statistics of Tweets for Word Embedding Training

Number of Tweets	198 million
Number of words in training data	2.9 billion
Number of unique tokens in the trained model	1.9 million

Each tweet used in the training process is preprocessed using steps similar to the ones for generating tf.idf value for a word, described in the evaluation subsection of section II, except the stop word removal step. Stop words are not removed for building the word embedding model, since they provide important context in which other words are used. These preprocessing steps are necessary, since most tokens removed or normalized are not useful, such as URLs, and keeping them will increase the vector space size and computing cost.

B. Data Set for Classification Evaluation

We collected 31,000 tweets belonging to the ten topic categories, and they were used as training and test data for

our evaluation. Each tweet was annotated and at least two annotators agreed on its label. These tweets were split into training, development, and test sets. Each tweet was preprocessed based on the preprocessing step described in previous sections.

C. Experiment and Result

Three tweet topic classifiers were built for this study based on features generated from these three types of data: tweet text, tweet text + paraphrases using WordNet, and tweet text + paraphrases using word embeddings. Given a tweet, the classifiers try to classify it into one of the predefined topic categories.

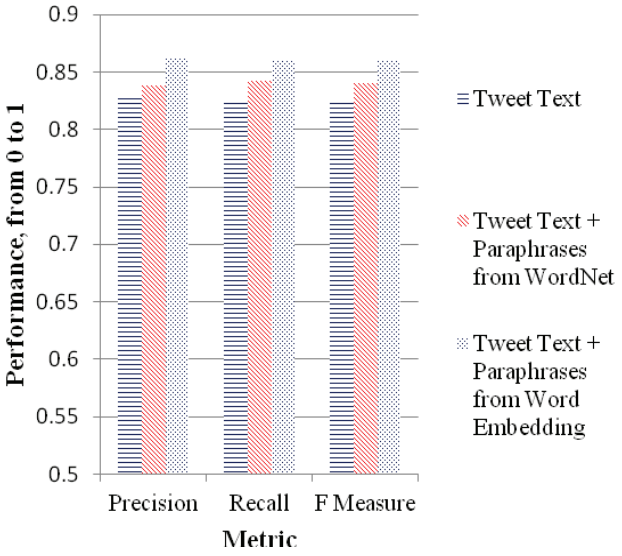

Figure 1. Tweet Topic Classification Performance Comparison Using the Three Approaches

Figure 1 presents the experiment result. We can see that both the classifier using WordNet for paraphrases and the one using word embeddings outperform the baseline approach, which does not use paraphrases. It also shows that the classifier using paraphrases built from word embeddings performs better than the one using WordNet. The performance differences are statistically significant at p=0.05 using t-test, for all the three classification metrics. There are several possible explanations that word embedding performs better than WordNet. One explanation is that the word embedding approach also uses paraphrases for hashtags and proper nouns, which the WordNet approach does not have. The second reason is that tweets have many terms and abbreviations that are not in a regular lexical database, such as WordNet, and a lexical database is usually not updated often enough to reflect the dynamics of social media. Another possible reason is that the paraphrases generated by the word embedding approach are not just synonyms, but they also include hypernomy, hyponomy, meronomy and holonomy, which means this approach provides more related terms than the synsets of WordNet. In tweet topic classification, the terms with these relations usually belong to the same topic category.

IV. Previous Studies

Several studies have attempted to overcome the data sparseness problem by exploiting external knowledge. One option is to use search engines to enrich the data context [1]. Another option is to use the embedded URL in a tweet to get the linked web page and use those pages to enrich the tweet data. Both approaches are not suitable for real-time applications. Another option is to use online data repositories, such as Freebase. Previous studies have used paraphrases in a number of tasks, such as machine translation, query expansion in a search engine, question answering, and event detection on social media [3, 8, 9].

Word embedding is a dense, low-dimensional and real-valued vector for a word, and it has been researched and used in related NLP tasks [10, 13]. Collobert et al. [4] introduce C&W model to learn word embedding based on the syntactic contexts of words. Another implementation is the word2vec from Mikolov et al. [5, 6], which is used to generate word embeddings in our study.

References

[1] Bollegala, D., Y. Matsuo, and M. Ishizuka. 2007, Measuring semantic similarity between words using Web search engines. Proc. WWW, 2007

[2] Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. NAACL 2006

[3] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. WWW 2006

[4] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. J. Mach. Learn. Res.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

[7] Christiane Fellbaum. 1998. WordNet: An electronic lexical database. The MIT press.

[8] S. Petrovic, M. Osborne, V. Lavrenko, Using paraphrases for improving first story detection in news and Twitter, NAACL 2012

[9] T. Kenter, M. Rijke, Short text similarity with word embeddings, CIKM 2015

[10] Richard Socher , Alex Perelygin , Jean Y. Wu , Jason Chuang , Christopher D. Manning , Andrew Y. Ng , Christopher Potts, Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. EMNLP 2013.

[11] O. Owoputi, B. O'Connor, C. Dyer, K. Gimpel, N. Schneider, and N. A. Smith. Improved part-of-speech tagging for online conversational text with word clusters. NAACL-HLT 2013.

[12] Christiane Fellbaum. Wordnet and wordnets. In Keith et al. Brown, editor, Encyclopedia of Language and Linguistics, Second Edition. Elsevier, Oxford, 2005

[13] Quanzhi Li, Sameena Shah, Xiaomo Liu, Armineh Nourbakhsh, Rui Fang, TweetSift: Tweet Topic Classification Based on Entity Knowledge Base and Topic Enhanced Word Embedding, CIKM 2016