Paraphrase Identification and Semantic Similarity in Twitter with Simple Features

Ngoc Phuoc An Vo
Fondazione Bruno Kessler,
University of Trento
Trento, Italy
ngoc@fbk.eu

Simone Magnolini
University of Brescia,
Fondazione Bruno Kessler
Trento, Italy
magnolini@fbk.eu

Octavian Popescu
IBM Research, T.J. Watson
Yorktown, US
o.popescu@us.ibm.com

Abstract

Paraphrase Identification and Semantic Similarity are two different yet well related tasks in NLP. There are many studies on these two tasks extensively on structured texts in the past. However, with the strong rise of social media data, studying these tasks on unstructured texts, particularly, social texts in Twitter is very interesting as it could be more complicated problems to deal with. We investigate and find a set of simple features which enables us to achieve very competitive performance on both tasks in Twitter data. Interestingly, we also confirm the significance of using word alignment techniques from evaluation metrics in machine translation in the overall performance of these tasks.

1 Introduction

Paraphrase Identification and Semantic Similarity are important tasks that can be used as features to improve many other Natural Language Processing (NLP) tasks, e.g. Information Retrieval, Machine Translation Evaluation, Text Summarization, Question and Answering, and others. Besides this, analyzing social media data like tweets of the social network Twitter is a field of growing interest for different purposes. The study of these typical NLP tasks on Twitter data can be very interesting as social media data carries lot of surprises and unpredictable information.

The Paraphrase Identification is a classic NLP task which is a classification problem. Given a pair of sentences, the system is required to assess if the two sentences carry the same meaning, to classify them *paraphrase*, or *not paraphrase* otherwise. Likewise, Semantic Similarity is another NLP task in which the system needs to examine the similarity degree (in a pre-defined semantic scale) of a given pair of texts, varying in different levels such as word, phrase, sentence, or paragraph.

There are different approaches, both supervised and unsupervised, have been proposed for these two tasks, ranging from simple level like word/n-gram overlapping, string matching, to more complicated ones like semantic word similarity, word alignment, syntactic structure, etc.[1,2] However, it is challenging or even inapplicable to deploy all these approaches to social media data, like Twitter data, due to many differences the social media data carries, such as misspelling, word out of vocabulary, slang, acronyms, style, structure, etc. In this paper, we study and find a set of simple features specifically chosen and suitable for social media data which is relatively easy to obtain, but able to achieve very competitive performance on both tasks for Twitter data. We also analyze the significance of each feature quantitatively and qualitatively in the overall performance. As a result, we can prove our hypothesis that the combination of simple features like word/n-gram overlapping, word alignment, and semantic word similarity can result in very good performance for both tasks on social media data.

The paper is organized as follows: Section 2

[1] http://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_(State_of_the_art)

[2] http://aclweb.org/aclwiki/index.php?title=Similarity_(State_of_the_art)

Proceedings of SocialNLP 2015@NAACL-HLT, pages 10–19,
Denver, Colorado, June 5, 2015.

presents the Related Work, Section 3 describes the tasks and set of features, Section 4 shows the Experiments, Section 5 reports the Evaluations, Section 6 discusses the Error Analysis, and finally Section 7 is the Conclusions and Future Work.

2 Related Work

The ability to identify paraphrase, in which a sentences express the same meaning of another one but with different words, has proven useful for a wide variety of natural language processing applications (Madnani and Dorr, 2010). The ACL Wiki gives an excellent summary of the state-of-the-art paraphrase identification techniques; this shows how much effort researchers did to automatically detecting paraphrases.[3] The different approaches can be categorized into supervised methods, i.e. (Madnani et al., 2012), (Socher et al., 2011) and (Wan et al., 2006), that, at the moment, are the most promising and unsupervised methods, i.e. (Fernando and Stevenson, 2008), (Hassan and Adviser-Mihalcea, 2011) and (Islam and Inkpen, 2009). Previous works use the Microsoft Research Paraphrase Corpus (MSRP) dataset (Dolan et al., 2004) that is obtained by extracting sentences from news sources on the web; however, this scenario is very different from social data. A few recent studies have highlighted the potentiality and importance of developing paraphrase (Zanzotto et al., 2011) and (Xu et al., 2013) and semantic similarity techniques (Guo and Diab, 2012) specifically for Tweets. They also indicated that the very informal language, especially the high degree of lexical variation, used in social media has posed serious challenges. Twitter data and, more in general, social media data have been used as dataset in a growing topic of research. Twitter, at the moment the most used microblogging tool, has seen a lot of growth since it launched in October, 2006. In (Java et al., 2007) preliminary analysis they find user clusters based on user intention to topics by clique percolation methods. This research is expanded and improved in several ways in (Krishnamurthy et al., 2008), they applied geographical characterization to cluster users and also found relation between the number of following and followers of a user. These and other similar researches have helped to obtain a more precise idea about some effect that action in this microblogging platform can have; (Kwak et al., 2010) use previous works as a base to rank users adding the effect of retweets on information propagation. With the data obtained from the population of blogs and social networks, opinion mining and sentiment analysis became, in the last years, a field of interest for many researches. In the literature (Pak and Paroubek, 2010), they describe a method for an automatic collection of a corpus that can be used to train a sentiment classifier. In a further research (Kouloumpis et al., 2011), it shows that part-of-speech features may not be useful for sentiment analysis in the microblogging domain, instead using hash-tags to collect training data did prove useful, as did using data collected based on positive and negative emoticons.

3 Paraphrase and Semantic Similarity in Twitter

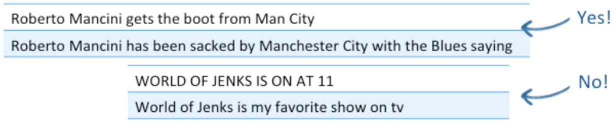
In this section, we introduce the two tasks Paraphrase Identification and Semantic Similarity in Twitter, then we describe the set of simple features which enables us to achieve competitive performance in both tasks.

3.1 Task Description

This is a shared-task proposed as the Task#1 "Paraphrase and Semantic Similarity in Twitter" at SemEval 2015 (Xu et al., 2015).[4] In this task, the first common ground for development and comparison of Paraphrase Identification (PI) and Semantic Similarity (SS) systems for the Twitter data is provided. Given a pair of sentences from Twitter trends, systems are required to produce a binary *yes/no* judgment and an optionally graded similarity score in the scale [0-1] to measure their semantic equivalence. This task is used to promote this line of research in the new challenging setting of social media data, and help to advance other NLP techniques for noisy user-generated text in the long run. Figure 1 shows examples of paraphrase and non-paraphrase pairs in Twitter.

[3] http://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_%28State_of_the_art%29

[4] http://alt.qcri.org/semeval2015/task1/


Roberto Mancini gets the boot from Man City

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

WORLD OF JENKS IS ON AT 11

World of Jenks is my favorite show on tv

Figure 1: Examples of PI in Twitter.

3.2 Data Preprocessing

In order to optimize the system performance, we carefully analyze the dataset and notice that Tweets' topic is a part that is always present in both sentences; this redundant similarity in the pairs does not give any information about paraphrase as two sentences always have the same topic, yet they may be paraphrase or not. Hence, we remove the topic from the sentences, and we did the same in the pairs with Part-of-Speech (POS) and named entity tags. As being suggested by the guideline of the task, we also remove all the pairs with uncertain judgment, such as "debatable" since they cannot confirm the *paraphrase/not paraphrase* relation between two sentences. After this data processing, we obtain two smaller datasets with very short texts, sometime reduced to a single word and with very poor syntactic structure. We split the original dataset into two subsets, in which one is composed by sentence pairs and the other one is composed by pairs with POS and named entity tags.

As Twitter data and other micro-blog data are usually informal text which is quite short in length and written in a variety of noise of presentation, e.g *"coooooool"* v.s *"cool"*, *"talkin"* v.s *"talking"*, *"u"* v.s *"you"*, *"thinkin"* v.s *"thinking"*, *"abt"* v.s *"about"*, etc. We apply the lexical normalization method (Han et al., 2013) to normalize noisy lexical from the input data. We also notice the simple structure of given datasets, especially, after undergoing the preprocessing, we decide to focus on exploiting the lexical and string similarity information, rather than syntactic information.

3.3 Feature Set

In order to build the system, we investigate and extract a set of simple features especially tailored for social media data which can be used for both tasks, for building either a binary classifier for detecting paraphrase or regression model to compute the similarity scores on Twitter data. Moreover, these features can be used independently or together with others to measure the semantic similarity and recognize the paraphrase of given sentence pair as well as to evaluate the significance of each feature to the accuracy of system's predictions. On top of this, the system is expandable and scalable for adopting more useful features aiming for improving the accuracy.

Lexical and String Similarity. We use the system described in the literature (Das and Smith, 2009) to compute the lexical and string similarity between two sentences by using a logistic regression model with eighteen features based on n-grams. This system uses precision, recall and F1-score of 1-gram, 2-gram and 3-gram of tokens and stems from sentence pair to build a binary classification model for identifying paraphrase. We extract the eighteen features from this system to use in our classification model.

Machine Translation Evaluation Metrics. Other than similarity features, we also use evaluation metrics in machine translation as suggested in (Madnani et al., 2012) for paraphrase recognition on Microsoft Research paraphrase corpus (MSRP) (Dolan et al., 2004). In machine translation, the evaluation metric scores the hypotheses by aligning them to one or more reference translations. We take into consideration to use all the eight metrics proposed, but we find that adding some of them without a careful process of training on the dataset may decrease the performance of the system. Thus, we only use two metrics in our system, the METEOR and BLEU. We actually also take into consideration the metric TERp (Snover et al., 2009), but it does not make any improvement on system performance, hence, we exclude it.

METEOR (Metric for Evaluation of Translation with Explicit ORdering). We use the latest version of METEOR (Denkowski and Lavie, 2014) that find alignments between sentences based on exact, stem, synonym and paraphrase matches between words and phrases. We used the system as distributed on its website using only the "norm" option that tokenizes and normalizes punctuation and lowercase as suggested by documentation.[5] We compute the word alignment scores on sentences and on sentences with part-of-speech and named entity tags, as our idea is

[5] http://www.cs.cmu.edu/ alavie/METEOR/index.html

Classifier/Features	Word/ n-grams Overlap (1)	(1) +METEOR	(1) +METEOR +TERp	(1) +METEOR +BLEU	(1) +METEOR +BLEU +EditDistance
Baseline-1	72.4	-	-	-	-
EditDistance	73.3	-	-	-	-
Decision Stump	73.7	74.4	74.4	74.4	74.4
OneR	73.7	74.4	74.4	74.4	74.4
Logistic	73.6	74.9	74.9	74.9	75.0
J48	72.6	74.7	74.2	74.6	74.7
BaysianLogisticRegression	72.0	74.9	74.8	74.9	75.0
VotedPerceptron	73.7	75.6	75.5	75.8	**76.2**
MultiLayerPerceptron	73.9	75.6	75.3	75.4	76.1

Table 1: Paraphrase Identification Accuracy (%) obtained using different classifiers with different features on Development data.

that if two sentences are similar, their tagged version also should be similar.

BLEU (Bilingual Evaluation Understudy). We use another metric for machine translation BLEU (Papineni et al., 2002) that is one of the most commonly used and because of that has an high reliability. It is computed as the amount of n-gram overlap, for different values of n=1,2,3, and 4, between the system output and the reference translation, in our case between sentence pairs. The score is tempered by a penalty for translations that might be too short. BLEU relies on exact matching and has no concept of synonymy or paraphrasing. As the length of tweets is relatively short, it is only 140-character message, we do not expect to have large n-gram overlaps, except 1-gram and 2-gram. Our analysis actually shows that 3-gram, 4-gram and the average score may cause more noise.

Edit Distance. We use the edit distance between sentences as a feature. For that we used the Excitement Open Platform (EOP) (Magnini et al., 2014).[6] To obtain the edit distance, we use EDITS Entailment Decision Algorithm (EDITS EDA) taking the edit distance instead of entailment or not entailment decision. We configure the system to use lemmas and synonyms as identical words to compute sentence distance, the system normalizes the score on the number of token of the shortest sentence. We choose this configuration because it returns the best performance evaluated on training and development data.

Sentiment Analysis. We speculate to improve paraphrase detection by adding a feature based on polarity given by a sentiment analysis system. We evaluate this feature on all three datasets (training, develpment, and testing). We use the Sentiment Pipeline of Stanford CoreNLP (Manning et al., 2014) to obtain this feature. We configure the pipeline for tokenizing, splitting sentence, POS tagging, lemmatization , parsing, named entity recognition (NER) and, of course, sentiment analysis. Despite the deep analysis, most of sentences are classified as either *"positive"*, *"negative"* or *"neutral"*; classes *"very positive"* and *"very negative"* are rare. We decide to use this as a polarity-matching feature (i.e. when both sentences in the pair are classified the same class), so we analyze the distribution of paraphrase and polarity matching on the three datasets, which results are shown in Table 2, Table 3 and Table 4. Contrary to our intuition, this feature seems not to be strongly correlated with paraphrasing, in particular, pairs with polarity matching have 2.08% more of probability to be paraphrase in the training dataset, a bit more (3.65%) in the development dataset, but even less (1.76%) in the test dataset. We also compute the information gain of the feature in the training dataset using WEKA (Hall et al., 2009) InfoGainAttributeEval with the default

[6] http://hltfbk.github.io/Excitement-Open-Platform/

ranker and we obtain a low result, only 0.00107, so we decide to exclude this approach. We still think that sentiment analysis could be an useful feature for paraphrase detection, and there would be a way to use it properly. To prove that, we try another different approach, instead of using a binary feature, we use three possible values: 0 if the polarity is opposite ("positive" and "negative"), 0.5 if one or both sentences in the pair are classified as "neutral" and 1 if they have the same polarity (both "positive" or "negative"). We compute the information gain of the feature in the training dataset and obtain a more promising score of 0.01272; this seems to confirm our idea on the sentiment analysis. Probably a wider range of values (more than just a 3 sub-classes) would possibly obtain better results. We aim to use a continuous value that describes polarity distance to improve our system performance.

	Paraphrase	Not Paraphrase
Without Sent. An.	3996 / 11530 34.66 %	7534 / 11530 65.34 %
Match	1856 / 5052 36.74 %	3196 / 5052 63.26 %
Mismatch	2140 / 6478 33.03 %	4338 / 6478 66.97 %

Table 2: Distribution of the paraphrase in training dataset without sentiment analysis and with polarity matching and mismatching.

	Paraphrase	Not Paraphrase
Without Sent. An.	1470 / 4142 35.49 %	2672 / 4142 64.51 %
Match	750 / 1916 39.14 %	1166 / 1916 60.86 %
Mismatch	720 / 2226 32.35 %	1506 / 2226 67.65 %

Table 3: Distribution of the paraphrase in development dataset without sentiment analysis and with polarity matching and mismatching.

	Paraphrase	Not Paraphrase
Without Sent. An.	175 / 838 20.88 %	663 / 838 79.12 %
Match	84 / 371 22.64 %	287 / 371 77.36 %
Mismatch	91 / 467 19.49 %	376 / 467 80.51 %

Table 4: Distribution of the paraphrase in test dataset without sentiment analysis and with polarity matching and mismatching.

3.4 Classification Algorithms

We build different models for both tasks using several widely-used classification algorithms (i.e. Decision Stump, OneR, Logistic, J48, BaysianLogisticRegression, VotedPerceptron, and MultiLayerPerceptron) to optimize 1) the Accuracy and F1-score for Paraphrase Identification and 2) the Pearson correlation of Semantic Similarity scores between system and human annotation. We use WEKA (Hall et al., 2009) to obtain robust and efficient implementation of the classifiers. We try several classification algorithms in WEKA, among others, we find that the VotedPerceptron classifier (*exponent 0.8*) returns the best result for the evaluation on training and development data. VotedPerceptron (Freund and Schapire, 1999) is a simple algorithm for linear classification which takes advantage of data that are linearly separable with large margins.

Classifier	F1-score
Baseline-1	0.502
EOP EditDistance	0.609
Decision Stump	0.736
OneR	0.733
Logistic	0.724
J48	0.721
BaysianLogisticRegression	0.723
VotedPerceptron	**0.746**
MultiLayerPerceptron	0.741

Table 5: Paraphrase Identification F1-score obtained using different classifiers on the best set of features (word/n-gram overlap + METEOR + BLEU + EditDistance).

METEOR(1)	BLEU(2)	EditDist(3)	WMF(4)	(1),(2)&(3)	(1),(2)&(4)	(2),(3)&(4)	All
0.4624	0.4022	0.4800	0.3304	**0.531**	0.471	0.515	0.526

Table 7: Semantic Similarity Results with different features on Test data.

Setup	Train&Dev	Test
Total (pairs)	18,000	972
Para	35%	32%
Non-Para	65%	68%
Selected	different trends	different times
Annotated by	5 AM Turkers	experts

Table 6: Distribution of Datasets.

4 Experiments

In this section, we describe the dataset, the task baselines and experiments carried on these two tasks.

4.1 Dataset

The dataset (Xu et al., 2014) consists of three parts, the training and development datasets (18,000 sentence pairs), the test dataset (972 sentence pairs) for evaluation. Table 6 presents the setup and distribution of all datasets used for the experiments.

Each row of data contains six tab-separated columns presenting the *Trending_Topic_Name, Sent_1, Sent_2, Label, Sent_1_tag* and *Sent_2_tag*. The *Sent_1* and *Sent_2* are two sentences which may not be necessarily full tweets. The *Label* column is in a format such like "(1, 4)", which means among 5 votes from Amazon Mechanical turkers only 1 is positive and 4 are negative. The mapping suggestions to binary labels are as follows:

- **paraphrases**: (3, 2) (4, 1) (5, 0)
- **non-paraphrases**: (1, 4) (0, 5)
- **debatable**: (2, 3) which may be discarded.

The *Sent1_tag* and *Sent2_tag* are the two sentences with part-of-speech and named entity tags. However, there is no labels of semantic similarity scores provided in development and training data, but only evaluation data.

4.2 Baselines

According to the task evaluation, we use all three baselines provided for this task which are placed at different advance levels.

Baseline-1 is a logistic regression model using simple lexical features, which is originally used in the literature (Das and Smith, 2009). It uses precision, recall and F1-score of 1-gram, 2-gram and 3-gram of tokens and stems from sentence pair to build a binary classification model for identifying paraphrase. This is the strongest baseline as it has the state-of-the-art level performance in the paraphrase identification literature.

Baseline-2 is the Weighted Matrix Factorization (WMF) model (Guo and Diab, 2012) which is a dimension reduction model to extract nuanced and robust latent vectors for short texts/sentences. To overcome the sparsity problem in short texts/sentences (e.g. 10 words on average), the missing words, a feature that LSA/LDA typically overlooks, is explicitly modeled. We use the pipeline to compute the similarity score between texts.[7]

Baseline-3 is a Random system which uses the *random* module in Python to generate a random score, in the scale [0 - 1], for each sentence pair, then it sets the threshold 0.5 for classifying *paraphrase* and *not paraphrase*.[8]

4.3 Paraphrase Identification

In order to optimize the Accuracy and F1-score for the classification, we build several models with different sets of features on the training data and evaluate these models on the development data to find the best feature set. The combination of word/n-gram, word alignment by METEOR, BLEU and EditDistance scores proves to be the most prominent set of simple features which can achieve very good performance. For classification algorithm, the VotedPerceptron returns the best result among other algorithms implemented in WEKA. In Table 1, we report the Accuracy results obtained by using different classifiers with different features. Our chosen classification algorithm and feature set outperform the strongest baseline and EOP EditDistance (standalone setting).

[7] http://www.cs.columbia.edu/%7Eweiwei/code.html

[8] https://docs.python.org/2/library/random.html

Table 5 shows F1-score obtained with different classifiers on our best set of features discovered in Table 1, and our system again results better than the strongest baseline and EOP EditDistance. Interestingly, the WMF feature which is expected to have some impact on computing the semantic similarity score does not incorporate well with other features.

4.4 Semantic Similarity

Due to no training data is given for computing the semantic similarity, a different approach is needed. Firstly, we consider to use external data from the similar task, which is Task #2 "Semantic Textual Similarity (STS)" (English STS) for training a semantic similarity model. However, after some preliminary experiments and analysis, we realize that this does not benefit our task on Twitter data due to the very big difference between formal text and informal text being used. We will need more study on how to use formal text to benefit informal text in the same task. Hence, we decide to build an unsupervised model for semantic similarity on Twitter data instead. We first adopt the result of Basline-2 (WMF) as a feature for semantic similarity. We build different unsupervised models which average the values of different sets of features learned for Paraphrase Identification task. Table 7 shows the Pearson correlation between the average of feature values and the gold similarity scores on the test data.

5 Evaluations

In this section, we discuss about the evaluation on both tasks. Table 8 shows the performance of our best models constructed by best sets of features in comparison with all the three baselines and the top three best systems reported in the shared-task.[9] For Paraphrase Identification task, our system outperforms all three baselines and achieves a very competitive result to the best systems. The difference between our system and the best three systems is a very small variance by a slim margin around 1%. In Semantic Similarity, though we only build simple model which averages the values of word alignment METEOR, BLEU and Edit Distance scores, our system still obtains better results than all three baselines and close to the top three results. These results on both tasks may place us at the 4th rank in comparison to the official ranking of the shared-task.

	PI			SS
System	Prec	Rec	F1	Pearson
Baseline-1	.679	.520	.589	.511
Baseline-2	.450	.663	.536	.350
Baseline-3	.192	.434	.266	.017
ASOBEK$^{(1^{st}\text{ PI})}$	.680	.669	.674	-
MITRE$^{(2^{nd}\text{ PI, }1^{st}\text{ SS})}$	.569	.806	.667	.619
ECNU$^{(3^{rd}\text{ PI})}$	.767	.583	.662	-
RTM-DCU$^{(2^{nd}\text{ SS})}$	-	-	-	.570
HLTC-UST$^{(3^{rd}\text{ SS})}$	-	-	-	.563
OurSystem	**.685**	**.634**	**.659**	**.531**

Table 8: Paraphrase Identification (PI) and Semantic Similarity (SS) Evaluation Results on Test data.

6 Error Analysis

In this section, we conduct an analysis of the misclassifications that our system makes on test data. We extract and show some randomly selected examples in which our system classifies incorrectly, both false positive or false negative; and then we analyze the possible causes for the misclassification. This inspection yields not only the top sources of error for our approach but also uncovers sources of unclear annotations in dataset.

True Positive	True Negative	False Positive	False Negative
111	612	51	64

Table 9: Error Analysis on Paraphrase Identification.

6.1 False Positive

[1357] *omg Family Guy is killing me right now - OMG we were quoting family guy*
[1357] *family guy is trending in the US - Family guy is so racist or maybe they just point out the racism in America*
[4135] *hahaha that sounds like me - That sounds totally reasonable to me*
[5211] *The world of jenks is such a real show - Jenks from the World of Jenks is such a good person*

[9] http://alt.qcri.org/semeval2015/task1/data/uploads/semeval-pit-2015-results.pdf

[128] *Anyone trying to see After Earth sometime soon - Me and my son went to see After Earth last night*

Though all these sentence pairs share many word similarity/matching and alignments, they are annotated as non-paraphrase. For example, the sentence pair **[4135]** has very high word matching and alignment after removing the common topic "sounds", but the important words "like" and "reasonable" which differ the meaning between two sentences, are not really semantically captured and distinguished by our system. As our system does not use any semantic feature, this kind of semantic difference is difficult to distinguish. Hence, it leads to false positive case.

6.2 False Negative

[4220] *Hell yeah Star Wars is on - Star Wars and lord of the rings on tv*
[785] *Chris Davis is putting the team on his back - Chris Davis doing what he does*
[400] *Rafa Benitez deserves a hell of a thank you - Any praise for Benitez from my Chelsea followers lol*
[2832] *Classy gesture by the Mets for Mariano - real class shown by The Mets Mo Rivera is a legend*
[4062] *Shonda is a freaking genius - THAT LADY IS AMAZING I LOVE SHONDA*

This case is opposite to the previous case, even though these sentence pairs do not share many word similarity and alignment, they are annotated as paraphrase. We can possibly propose some hypothesis as follows:

Extra information Though the pairs **[4220]** and **[400]** may not be paraphrase according to the paraphrase definition in the literature (Bhagat and Hovy, 2013), they are annotated as paraphrase in the gold-standard labels. In this case, we notice that as one sentence contains more extra information than the other one, it leads to low word similarity and alignment, which makes our system make wrong classification.

Specific knowledge-base In this case, the pairs **[785]** and **[2832]** require a specific knowledge-base, which is about baseball, to recognize the paraphrase; hence, even for human without any related knowledge, it might be difficult detect the paraphrase.

Common sense Though both sentences of the pair **[4062]** do not share any word similarity/alignment, they have a positive polarity that may allow identifying the paraphrase. This case may be easy for human to identify the paraphrase, yet it is difficult for machine to capture the same perception.

Table 9 shows that we can improve our system performance by exploiting more semantic features to make correct classification. Though we try to adopt the WMF which is supposed to provide more semantic information, it does not show any contribution in the overall performance. Moreover, according to our analysis for the false negative, it is rather difficult to cover these cases.

7 Conclusions and Future Work

In this paper, we study and present a set of simple features which is especially tailored to obtain very competitive performance in Paraphrase Identification and Semantic Similarity tasks on Twitter data. From the evaluation results, we can confirm our hypothesis in which the combination of word/n-grams overlap, METEOR word alignment, BLEU and Edit Distance scores can be an alternative approach to explore semantic information on Twitter data at a low cost. However, for future work, we expect to study more useful features (e.g the POS information, semantic word similarity) to improve the system performance on both identifying paraphrase and computing semantic similarity scores. From our error analysis, we consider to have more study on exploiting the semantic information for the task Semantic Similarity; and investigating on domain adaptation techniques for broad-topic data to benefit the task Paraphrase Identification in Twitter. Finally, we speculate the sentiment feature which seems to be promising in paraphrase identification task. More investigation and analysis will be needed for exploiting and integrating it with other features for better performance.

References

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

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. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, pages 45–52. Citeseer.

Yoav Freund and Robert E Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalisation of short text messages. *ACM Transactions on Intelligent Systems and Technology*, 4(1):5.

Samer Hassan and Rada Adviser-Mihalcea. 2011. *Measuring semantic relatedness using salient encyclopedic concepts*. University of North Texas.

Aminul Islam and Diana Inkpen. 2009. Semantic similarity of short texts. *Recent Advances in Natural Language Processing V*, 309:227–236.

Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. 2007. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM.

Efthymios Kouloumpis, Theresa Wilson, and Johanna Moore. 2011. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538–541.

Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. 2008. A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, pages 19–24. ACM.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM.

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 the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190. Association for Computational Linguistics.

Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Günter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. 2014. The excitement open platform for textual inferences.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 1320–1326.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127.

Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

Stephen Wan, Mark Dras, Robert Dale, and Cécile Paris. 2006. Using dependency-based features to take the "para-farce" out of paraphrase. In *Proceedings of the Australasian Language Technology Workshop*, volume 2006.

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. Citeseer.

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. Association for Computational Linguistics.