ASOBEK: Twitter Paraphrase Identification with Simple Overlap Features and SVMs

Asli Eyecioglu and **Bill Keller**
Department of Informatics
The University of Sussex
Brighton, UK
A.Eyecioglu@sussex.ac.uk,
billk@sussex.ac.uk

Abstract

We present an approach to identifying Twitter paraphrases using simple lexical overlap features. The work is part of ongoing research into the applicability of *knowledge-lean* techniques to paraphrase identification. We utilize features based on overlap of word and character n-grams and train support vector machine (SVM). Our results demonstrate that character and word level overlap features in combination can give performance comparable to methods employing more sophisticated NLP processing tools and external resources. We achieve the highest F-score for identifying paraphrases on the Twitter Paraphrase Corpus as part of the SemEval-2015 Task1.

1 Introduction

This paper presents an approach to identifying Twitter paraphrase pairs using lexical overlap features. Paraphrase identification (PI) may be defined as “the task of deciding whether two given text fragments have the same meaning” (Lintean & Rus 2011). Methods for identifying paraphrases thus take a pair of texts and make a binary judgment. The PI task has practical importance in the Natural Language Processing (NLP) community because of the pervasive problem of linguistic variation. Accurate methods for PI should help improve the performance of NLP systems that would seem to require language understanding. This includes key applications such as question answering, information retrieval and machine translation, amongst others. Acquired paraphrases have been shown to improve the performance of Statistical Machine Translation (SMT) systems, for example (Callison-Burch et al. 2006, Owczarzak et al., 2006; Madnani et al., 2007)

Many researchers on PI make use of existing NLP tools and other resources to identify paraphrases. For example, Duclaye et al., (2002) exploits the NLP tools of a question answering system for reformulating rules to identify paraphrases. Other researchers (Finch et al 2005, Mihalcea et al 2006, Fernando & Stevenson 2008, Malakasiotis 2009, Das & Smith 2009) have employed lexical semantic similarity information based on resources such as WordNet (Miller, 1995).

Although the PI task aims to identify sentences that are semantically equivalent, a number of researchers have shown that classifiers trained on lexical overlap features may achieve relatively high accuracy. Good performance is achieved without the use of knowledge-based semantic features or other external knowledge sources such as parallel corpora (Lintean & Rus 2011, Blacoe & Lapata, 2012). We consider methods as *knowledge-lean* if they make use of just the text at hand and avoid the use of external processing tools and other resources. Knowledge-lean PI methods may thus employ shallow overlap measures based on lexical items or n-grams, but they might also make use of distributional techniques where these are based on simple text statistics.

The work described here is part of ongoing research that is investigating the extent to which knowledge-lean techniques may help to identify paraphrases. Preliminary work has been conducted using the Microsoft Research Paraphrase Corpus

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 64–69,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

(MSRPC) (Dolan & Brockett, 2005). However, the approach may be of particular value where knowledge-based language resources are not readily available or applicable. In this context, Twitter presents interesting challenges. Its short texts (tweets), widespread use of non-standard grammar, spelling and punctuation, as well as slang, abbreviations and neologisms, etc. make syntactic and semantic analysis difficult.

We apply a supervised learning approach using SVMs and learn classifiers based on simple lexical and character n-gram overlap features. SVM classifiers benefit from features that are interdependent and informative, so good choice of feature combinations is crucial. We also experimented with different kernels to find out whether a non-linear kernel works well for this task.

2 Related Work

A number of researchers have investigated whether near state-of-the-art PI results can be obtained without use of external sources. Blacoe & Lapata (2012) use distributional methods to find compositional meaning of phrases and sentences. They find that performance of shallow approaches is comparable to methods that are computationally intensive or that use very large corpora. Lintean & Rus (2011) apply word unigrams and bigrams. Bigrams capture word order information, which can in turn capture syntactic similarities between two text fragments. Finch et al. (2005) combines several MT metrics and uses them as features. Madnani et al. (2012) also shows that good results are obtained by combining different MT metrics. Ji & Eisenstein (2013) attains state-of-the-art results based on latent semantic analysis and a new term-weighting metric, TF-KLD.[1]

A variety of classifiers has been employed for the purpose of identifying paraphrases. Kozarova & Montoyo (2006) measures lexical and semantic similarity with the combination of different classifiers: k-Nearest Neighbours, Support Vector Machines, and Maximum Entropy. The SVM Classifiers remains the most applicable in recent research whether applied solely (Finch et al., 2005; Wan et al., 2006) or part of combined classifiers (Kozoreva & Montotyo, 2006; Lintean & Rus, 2011; Madnani et al, 2012).

3 The Task

The Semeval-2015 task “Paraphrase and Semantic Similarity in Twitter” involves predicting whether two tweets have the same meaning. Training and test data are provided in the form of a Twitter Paraphrase Corpus (TPC) (Xu, 2014). The TPC is constructed semi-randomly and annotated via Amazon Mechanical Turk by 5 annotators. It consists of around 35% paraphrases and 65% non-paraphrases. Training and development data consists of 18K tweet pairs and 1K test data. Test data is drawn from a different time period and annotated by an expert.

4 Approach

4.1 Text Preprocessing

Text preprocessing is essential to many NLP applications. It may involve tokenizing, removal of punctuation, PoS-tagging, and so on. For identifying paraphrases, this may not always be appropriate. Removing punctuation and stop words, as commonly done for many NLP applications, arguably results in the loss of information that may be critical in terms of PI. We therefor keep text preprocessing to a minimum.

The TPC is already tokenized (O’Connor et al., 2010), part-of-speech tagged (Derczynski et al., 2013), and named entity tagged (Ritter et al., 2011). Here we only experiment on tokenized data, ignoring part-of-speech and named entity tagged data. In the next section we also report results for the MSR Paraphrase Corpus. We used the Rasp Toolkit (Briscoe et al., 2006) to perform tokenization in this case.

A particular issue in dealing with Twitter is the use of capitalization. Variability in the use of capitals (some tweets may be uncapitalised, others written in all uppercase) presents a problem for simple lexical overlap measures between candidate paraphrase pairs. To help overcome this, tokenized tweets are lowercased. Although this potentially causes confusion between proper nouns and common nouns (e.g. *apple* the fruit v. *Apple* the company) our experimental work shows that it most likely increases the quantity of identified paraphrase pairs.

Tweets tend to have a higher proportion of out-of-vocabulary (OOV) words than other texts.

[1] State-of-the-art results are shown in Section 5.

Due to the character limit, words are often shortened or abbreviated and standard spelling rules ignored. In addition, characters may be added for emphasis. Nevertheless, we have not normalized the original texts to compensate for this.

A novel aspect of the TPC compared to other paraphrase corpora is the inclusion of topic information, which is also used during the construction process. Despite the possibility that topic features might be utilized, we have not made use of this information in our approach.

4.2 Features and Instances

As the basis for deriving a number of overlap features, we consider different representations of a text as a set of tokens, where a token may be either a word or character n-gram. For the work described here we restrict attention to word and character unigrams and bigrams. Use of a variety of machine translation techniques (Madnani et al., 2012) that utilise word n-grams motivated their use in representing texts for this task. In particular, word bigrams may provide potentially useful syntactic information about a text. Character bigrams, on the other hand, allow us to capture similarity between related word forms. Possible overlap features are constructed using basic set operations:

Size of union: the size of the union of the tokens in the two texts of a candidate paraphrase pair.

Size of intersection: the number of tokens common to the texts of a candidate paraphrase pair.

Text Size: the size of the set of tokens representing a given text.

This yields a total of eight possible overlap features for a pair of texts, plus four ways of measuring text size. Each data instance is a vector of features representing a pair of tweets. In order to select an optimal set of features we ran a number of preliminary experiments. Table 1 presents the results for different features and combinations of features on the development data. We present results obtained for a linear kernel. The general pattern for an RBF kernel is similar.

Intuitively, knowing about the union, intersection or size of a text in isolation may not be very informative. However, for a given token type, these four features in combination provide potentially useful information about similarity of texts. In the following, C1 and C2 each denote four features (union, intersection, sizes of tweet 1 and tweet 2) produced by character unigrams and bigrams, respectively. Similarly, W1 and W2 denote the four features generated by word unigrams and bigrams, respectively. Combinations (e.g. C1W2) represent eight features: those for C1 plus those for W2.

Features	**Acc**	**Pre.**	**Rec.**	**F-sc.**
C1	64.5	0.0	0.0	0.0
C2	74.5	70.2	48.4	57.7
C1C2	74.5	70.3	48.5	57.4
W1	74.1	70.5	46.5	56.0
W2	70.5	63.9	38.8	48.3
W1W2	74.0	69.9	46.9	56.2
C1W1	74.2	70.4	47.2	56.5
C2W2	74.9	71.1	49.1	58.1
C1W2	71.4	72.0	31.9	44.2
C2W1	75.6	72.4	50.6	59.6
Baseline	72.6	70.4	38.9	50.1

Table 1: Individual and Combined Results by Linear SVM

It is clear that features based on character bigrams are more informative than character unigrams (for C1, all instances are classified negative). For words, on the other hand, use of bigrams did not improve performance over unigrams. However, combining features for words and characters proved beneficial. Although, the combination of character and word bigrams increases performance, combining word unigrams and character bigrams is more informative. We therefore chose to represent instances using a combination of character bigrams and word unigrams. [2]

An important step in SVM classification is rescaling of the features. Apart from a simple scaling mechanism, which is applied during the classification process, features are kept as they are.

4.3 SVM Classifiers

An SVM classifier maps the feature vectors into high dimensional vector space and computes the dot product of the two vectors inside the kernel. Its applicability to both linear and non-linear systems has been proven for different NLP applications. We used SVM implementations from scikit-learn (Pedregosa et al., 2011) and experimented with a number of classifiers. We report here on results obtained using SVC adapted from libsvm (Chang & Lin, 2011) by embedding different kernels. We

[2] The submitted system used just six features: four character bigram features together with just the union and intersection of word unigrams. This had no impact on performance.

experimented with linear and Radial Basis Function (RBF) kernels. Linear kernels are known to work well with large datasets and RBF kernels are the first choice if small number of features are applied (Hsu et al., 2003), which both cases to apply our datasets. Classifiers are used with their default parameters and trained on the data provided.

5 Results

Table 2 shows that SVC with a linear kernel achieved an F-score of 67.4. This represent the highest score amongst those systems participating in Task 1, though still some way below Xu et al (2014) and the human upper-bound. Xu et al. (2014)'s approach constructs a joint word-sentence paraphrase model (MULTIP) and utilizes topic information, which outperforms other features individually. Table 2 also shows the result for the RBF kernel, which was not submitted for the task. For this task the non-linear kernel does not provide any performance gain over the linear SVM.

Model	**Acc.**	**Pre.**	**Rec.**	**F-sc.**
Human Upperbound	--	75.2	90.8	82.3
Xu et al. (2014)	--	72.2	72.6	72.4
SVC (linear kernel)	86.5	68.0	66.9	67.4
SVC (rbf kernel)	85.7	64.9	68.6	66.7
Baseline	--	67.9	52.0	58.9

Table 2: TPC Results

For comparison, Table 3 shows state-of-the-art results for the PI task on the MSRPC, together with our classifiers trained using of same set of features as for the TPC. Our method performs well above baseline, but with relatively lower precision than other systems. In contrast to Table 2, our highest result is obtained using the RBF kernel.

Model	**Acc.**	**Pre.**	**Rec.**	**F-sc.**
Ji&Eisenstein(2013)	80.4			85.96
Madnani et al (2012)	77.4	-	-	84.1
Socher et al. (2011)	76.8	-	-	83.6
Wan et al. (2006)	75.6	77.0	90.0	83.0
SVC(rbf kernel)	**74.4**	**74.8**	**92.9**	**82.8**
Das & Smith (2009)	76.1	79.6	86.1	82.7
Finch et al. (2005)	75.0	76.6	89.8	82.7
Fernando&Stevenson (2008)	74.1	75.2	91.3	82.4
SVC (linear kernel)	**73.7**	**75.0**	**90.1**	**82.1**
Qiu et al. (2006)	72.0	72.5	93.4	81.6
Zhang and Patrick (2005)	71.9	74.3	88.2	80.7
BASELINE	**65.4**	**71.6**	**79.5**	**75.3**

Table 3: Paraphrase Identification State-of-the-art Results on MSRPC

We note that the features that we adopt as informative for the Twitter PI task outperform some recent approaches to PI on the MSRPC. This is encouraging and indicates applicability of knowledge-lean approaches to other data sets.

6 Conclusions

Our results demonstrate that knowledge-lean methods based on character and word level overlap features in combination can give good results in terms of the identification of Twitter paraphrases. SVM classifiers were successfully used to identify paraphrase pairs given just a few simple features. Our approach performed as well as and generally much better (in terms of F-score) than other, more sophisticated participating systems.

Overlap of character bigrams was more informative than that of character unigrams. We hypothesize that measuring overlap of character bigrams provides a way of detecting similarity of related word-forms. It thus performs a similar function to stemming or lemmatization in other domains, whilst retaining some information about difference. This may be especially helpful with Twitter, where a variety of idiosyncratic spellings and short forms may be observed alongside the usual morphological variants.

A strength of our approach is that preprocessing is kept to a minimum. This may explain why our system outperforms other approaches that use a similar set of overlap features. Methods that require the removal of stop words, punctuation, OOV words etc., lose potentially useful information. On the other hand, we found that normalizing tweets with regard to capitalization enhanced performance of the classifier.

The current work on paraphrase identification is ongoing. There is clearly room for reaching to human upper bound shown in Table 2. Our latest work shows that extending character and word n-grams to up to 4 is promising and gives performance that is close to the state–of-the-art results on TPC obtained by Xu et al. (2014). We intend to report on these results in a future paper.

References

Blacoe, W., & Lapata, M. (2012). A Comparison of Vector-based Representations for Semantic Composition. *Proceedings of the 2012 Joint Confer-*

ence on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12), (July), 546–556.

Briscoe, E., J. Carroll and R. Watson (2006) The Second Release of the RASP System. *In Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions.* Sydney, Australia, 77-80.

Callison-burch, C., Koehn, P., & Osborne, M. (2006). Improved Statistical Machine Translation Using Paraphrases. *In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL '06). Association for Computational Linguistics* (pp. 17-24). Stroudsburg, PA, USA

Chang, C.C. & Lin, C.J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27.* Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Das, D., & Smith, N. A. (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 (ACL '09), Vol. 1. Association for Computational Linguistics* (pp. 468-476). Stroudsburg, PA, USA.

Derczynski, L., Ritter, A., Clark, S., & Bontcheva, K. (2013). Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data. *In Proceedings of the International Conference on Recent Advances in Natural Language Processing.*

Dolan, W. B. & Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases, 9-16. *In Proceedings of The Third International Workshop on Paraphrasing (IWP2005), Jeju, Republic of Korea.*

Duclaye, F., Yvon, F., Collin, O., R, F. T., Marzin, P., & Cedex, L. (2002). Using the Web as a Linguistic Resource for Learning Reformulations Automatically. *In Proceedings of the Third International Conference on Language Resources and Evaluation. (pp. 390-396).*

Fernando, S., & Stevenson, M. (2008). A Semantic Similarity Approach to Paraphrase Detection. In *In Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics* (pp. 45–52).

Finch, A., Hwang, Y.-S., & Sumita, E. (2005). Using Machine Translation Evaluation Techniques to Determine Sentence-level Semantic Equivalence. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)* (pp. 17–24).

C.-W. Hsu, C.-C. Chang, C.-J. Lin. (2003) A practical guide to support vector classification. *Technical report, Department of Computer Science, National Taiwan University. July.*

Ji, Y., & Eisenstein, J. (2013). Discriminative Improvements to Distributional Sentence Similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 891–896). Seattle, Washington, USA: Association for Computational Linguistics.

Kozareva, Z., & Montoyo, A. (2006). Paraphrase Identification on the Basis of Supervised Machine Learning Techniques. *in Proceedings of the 5th International Conference on Natural Language Processing (FinTAL 2006),Lecture Notes in Artificial Intelligence* (pp. 524-533). Turku, Finland.

Lintean, M., & Rus, V. (2011). Dissimilarity Kernels for Paraphrase Identification. *Proceedings of the 24th International Florida Artificial Intelligence Research Society Conference.* (pp. 263-268). Palm Beach, FL.

Madnani, N., Ayan, N. F., Resnik, P., Dorr, B. J., & Park, C. (2007). Using Paraphrases for Parameter Tuning in Statistical Machine Translation. *Proceedings of the Second Workshop on Statistical Machine Translation (WMT'07).* (Vol. 20742). Prague, Czech Republic: Association for Computational Linguistics.

Madnani, N., Tetreault, J., & Chodorow, M. (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 (NAACL HLT '12) (pp. 182-190).* Stroudsburg, PA, USA

Malakasiotis, P. (2009). Paraphrase Recognition Using Machine Learning to Combine Similarity Measures. *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop* (pp. 27-35). Suntec, Singapore.

Mihalcea, R., Corley, C., & Strapparava, C. (2006). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In A. Cohn (Ed.), *Proceedings of the 21st national conference on Artificial intelligence- Volume 1* (pp. 775–780). AAAI Press.

Miller, G.A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM Vol. 38, No. 11: 39:41.*

O'Connor, B., Krieger, M., & Ahn, D. (2010). TweetMotif : Exploratory Search and Topic Summarization for Twitter. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (pp. 384–385). Association for the Advancement of Artificial Intelligence.

Owczarzak, K., Gorves, D., Genabith, J. V., & Way, A. (2006). Contextual Bitext-Derived Paraphrases in Automatic MT Evaluation. *In Proceedings of the Workshop on Statistical Machine Translation (StatMT '06). Association for Computational Linguistics* (pp. 86-93). Stroudsburg, PA, USA.

Pedregosa, F., Varoquaux, G., Gramfort, A. , Michel, V.,Thirion, B.,Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research,* 12*, 2825-2830*

Qiu, L., Kan, M.-Y., & Chua, T.-S. (2006). Paraphrase Recognition via Dissimilarity Significance Classification. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*, (July), 18–26. doi:10.3115/1610075.1610079

Ritter, A., Clark, S., Mausam & Etzioni, O. (2011). Named entity recognition in tweets: an experimental study. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11). Association for Computitional Linguistics.* Stroudsburg, PA, USA, 1524-1534.

Socher, R., Huang, E. H., Pennington, J., Ng, A. Y., & Manning, C. D. (2011) *Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection*. *Science*, 1-9.

Wan, S., Dras, M., & Dale, R. (2006). Using Dependency-Based Features to Take the " Para-farce " out of Paraphrase. *In proceedings of the Australasian Language Technology Workshop* (pp. 131-138). Sydney, Australia.

Xu, W. (2014). Data-Drive Aprroches for Paraphrasing Across Language Variations. *PhD Thesis.* Department of Computer Science, New York University.

Xu, W., Ritter, A., Callison-Burch, C., Dolan, W., & Ji, Y. (2014). Extracting Lexically Divergent Paraphrases from Twitter. Transactions Of The Association For Computational Linguistics, 2, 435-448. Retrieved fromhttps://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/498

Zhang, Y., & Patrick, J. (2005). Paraphrase Identification by Text Canonicalization. *In proceedings of the Australasian Language Technology Workshop* (pp. 160-166). Sydney, Australia.