Paraphrase Identification and Applications in Finding Answers in FAQ Databases

António Amaral
Instituto Superior Técnico and INESC-ID
Rua Alves Redol, 9
1000-029 Lisboa, Portugal
antonio.amaral@ist.utl.pt

Abstract

This work revisits the task of identifying paraphrases (i.e., given a pair of sentences, classify them as being paraphrases or not paraphrases). We propose to address the task through supervised machine learning, training classification models based on ensembles of trees, which use features that mostly correspond to string similarity metrics relying on lexical information. The most innovative contributions of this work relate to (i) the usage of classification methods based on tree ensembles, (ii) the combination of machine translation metrics with other types of features, and (iii) the usage of similarity features based on distributional word clustering. We report on a set of experiments that used the well-known Microsoft Research Paraphrase Corpus, in which we achieved a classification accuracy of 0.77, and an F1 measure of 0.84. We therefore show that out-of-the-box learning algorithms and relatively simple features can obtain state-of-the-art results in this task.

1 Introduction

The act of paraphrasing is generally defined as the restatement (or reuse) of text, giving the same meaning in another form. The problem of modeling paraphrase relationships between natural language utterances has recently attracted significant interest. For computational linguists, solving this problem may shed light on how to best model the semantics of sentences. For natural language engineers, the task has important applications in different types of real-world problems that involve measuring semantic overlap between sentences, like abstractive summarizers, question answering and FAQ retrieval systems, machine translation systems, or systems for the automatic identification of copyright infringement.

In brief, we have that the paraphrase identification problem asks whether two sentences have essentially the same meaning. Although paraphrase identification is defined in semantic terms, it is often solved using statistical classifiers based on shallow lexical and n-gram overlap features (e.g., typical methods rely on lexical matching techniques, in which the similarity between two candidate texts is computed as a function of the number of matching sequences of tokens between the texts). Recently, authors have also experimented with the usage of supervised machine learning methods for combining multiple types of features, including word similarity information derived from WordNet or from distributional word clustering methods, or even syntactic features based on dependency parse trees. However, although the introduction of linguistically-informed features has been shown to boost performance, there are also drawbacks in terms of computational efficiency, and in terms of the required resources and linguistic processing. We have, for instance, that most previous work on the area has focused on the English language, using the benchmark provided by the Microsoft Research Paraphrase Corpus. However, applications to other languages would probably not be able to rely on linguistic resources similar to WordNet, or on robust NLP pipelines for extracting syntactic features, given that these may not be available.

In this paper, we revisit the task of paraphrase identification, modeling the problem as a classification task that we propose to address through models based on ensembles of trees, using a large set of features that essentially correspond to string similarity metrics relying on lexical information. The most innovative contributions of this work relate to (i) the usage of classification methods based on tree ensembles, (ii) the combination of machine

translation metrics with other types of features, and (iii) the usage of similarity features based on distributional word clustering. We argue that robust classification methods, together with features derived from word clustering and from machine translation metrics, can be enough for attaining good results, without requiring lexical resources like WordNet, or features based on syntactic parsing. We report on a set of experiments that used the well-known Microsoft Research Paraphrase Corpus (MSRPC), in which we achieved a classification accuracy of 0.77, and an F1 measure of 0.84. We therefore showed empirically that out-of-the-box learning algorithms and relatively simple features can obtain state-of-the-art results in this increasingly relevant task.

The rest of this paper is organized as follows: Section 2 presents related work in the area. Section 3 presents the proposed method, detailing for instance the considered features. Section 4 presents our experimental validation results. Finally, Section 5 presents our conclusions, and points possible directions for future work.

2 Related Work

Paraphrase identification is the task of deciding whether two text fragments, usually sentences, are paraphrases of each other. This task should not be confused with that of paraphrase extraction, which concerns with collecting pairs of text fragments that are paraphrases of each other, from sources such as the Web (Dolan et al., 2004).

Possibly the simplest approach to paraphrase identification is an information retrieval (IR) based bag-of-words strategy. This strategy calculates a cosine similarity score for the given sentence pair, and if the similarity exceeds a threshold (either empirically determined, or learned from supervised training data), the sentences are considered paraphrases. More complex approaches typically involve the combination of multiple similarity metrics, either though an heuristic method or through supervised learning.

We have for instance that Kozareva and Montoyo (2006) proposed a classification-based approach to sentence-level paraphrase identification, combining lexical and semantic similarity features, some of them based on well known text summarization measures. The experiments revealed that simple features relying on common consecutive or insequence matches can resolve a large number of paraphrases correctly, and that combining multiple classifiers can also be beneficial. Zia and Wasif (2012) proposed a similar method using an enhanced feature set, also leveraging semantic heuristics (i.e., negation patterns) to aid in the detection of false paraphrases.

Zhang and Patrick (2005) proposed another classification-based method, in which the source sentence pairs are first converted into surface text that approximates canonical forms, through a limited set of canonicalization rules (e.g., changing sentences from the passive to active voice, or replacing different types of numeric quantities by common tokens). A decision tree learning module, which employs simple lexical matching features (e.g., the edit distance between tokens in the canonical versions of the sentences, which is similar to the WER metric from machine translation), takes the output canonicalized texts as its input for the supervised learning process.

Finch et al. (2005) proposed a similar method to those of Zhang and Patrick (2005) or of Kozareva and Montoyo (2006), but instead relying on standard evaluation methods for Machine Translation (MT), which are based on n-gram overlaps or edit distances (e.g., BLEU, NIST, WER and PER) as the similarity metrics between the sentences, and using a Support Vector Machine (SVM) classifier with radial basis function kernels to classify the data. The authors used stemming to conflate morphologically related words (i.e., verbs and adjectives) to the same root, as a pre-processing step to canonicalize the sentences. The authors also introduced a method based on the PER MT evaluation metric, which leverages part-of-speech information of the words contributing to the word matches and non-matches in the sentence. More recently, Madnani et al. (2012) proposed yet another approach based on evaluation methods from the area of machine translation, in light of recent developments. These authors showed that a meta-classifier (i.e., a classifier that uses the average of the unweighted probability estimates given by three constituent classifiers to make the final decision, in which the constituent classifiers are based on logistic regression, support vector machines, and nearest-neighbor searching) using nothing but modern MT metrics (i.e., 8 different MT metrics, including 6 that were not available in 2005) outperforms many recent paraphrase identification approaches relying on more

linguistically-informed features.

The approach developed by Qiu et al. (2006) instead uses a two-phase process, also based on supervised classification. Unlike most paraphrase identification systems that focus on sentence similarity, Qiu et al. (2006) propose to detect dissimilarities between sentences, making the paraphrase judgment based on the significance of such dissimilarities. A first phase identifies the common information nuggets or key semantic content units in each sentence, pairing them between the sentences. These nuggets are predicate argument tuples (i.e., structured representations of a verb predicate together with its arguments, obtained from a semantic role labeler), which are compared using a lexical matching technique. In the second phase, any unpaired nuggets are classified by an SVM model as significant or not. This SVM classifier uses a wide set of features representing the unpaired tuples, including internal counts of numeric expressions, named entities, words, semantic roles, and whether they are similar to other tuples in the same sentence, as well as contextual features like source/target sentence length and paired tuple count. If the sentences do not contain unpaired nuggets, or if all unpaired nuggets are classified as insignificant, then the sentences are considered paraphrases.

Several authors have noted that paraphrases often involve the usage of synonyms or other forms of related words and, as such, paraphrase identification methods should also be informed by word-level similarity information. Islam and Inkpen (2007) have, for instance, used a corpus based measure of semantic word similarity (i.e., an approach based on the second order co-occurrence pointwise mutual information, previously described by Islam and Inkpen (2006)), together with normalized and modified versions of the Longest Common Subsequence (LCS) string matching algorithm (i.e., three different modified versions of LCS, taking a weighted sum of these scores). A supervised learning procedure was used to define the threshold over which sentences were considered to be paraphrases.

Mihalcea et al. (2006) and Fernando and Stevenson (2008) have both presented algorithms for paraphrase identification that make use of word similarity information, derived from WordNet, in the considered sentence similarity metrics. Mihalcea et al. (2006) used word-to-word similarity measures (e.g., knowledge-based metrics which use WordNet, or other corpus-based measures, namely approaches based on pointwise mutual information and latent semantic analysis) together with a word specificity measure to estimate the semantic similarity of sentence pairs. An heuristic function is used to combine both metrics, aligning pairs of words according to their maximal similarity, and weighting aligned words according to their specificity. A threshold value of 0.5 was finally used for classification (i.e., sentence pairs with a score above the threshold were classified as being paraphrases). Fernando and Stevenson (2008) instead proposed a matrix similarity approach in which all word-to-word similarities are taken into account, and not just the maximal similarities between the sentences, as in the method by Mihalcea et al. (2006). They experimented with six different WordNet similarity metrics to populate a similarity matrix, which is then used to compute the similarity between vectorial representations of the sentences.

Rus et al. (2008) proposed a paraphrase identification method based on lexico-syntactic graph subsumption, relying on lexical, syntactic, synonymy and antonymy information. The synonymy and antonymy information is extracted from WordNet. In this approach, graphs are used to model the linguistic information embedded in both sentences from a given pair, with vertices representing concepts and edges representing syntactic relations among concepts. The two sentences are paraphrases if each of the hypothesis graphs subsumes the other. The subsumption algorithm starts by finding an isomorphism between the sets of vertices associated to both sentences, using WordNet to find possible synonyms. Then, the algorithm checks whether the labeled edges also have correspondences, considering relation equivalences among linguistic phenomena such as possessives or negations (e.g., using WordNet antonyms to deal with negations). The final subsumption score is a weighted sum of each individual vertex and edge matching score, with the weights discovered through a linear regression procedure.

Socher et al. (2011) proposed an approach that incorporates the similarities between both single word features and multi-word phrases extracted from the nodes of parse trees. This approach involved two main components, namely (i) an un-

folding recursive auto-encoder for unsupervised feature learning from unlabeled parse trees, and (ii) a dynamic pooling layer which outputs a fixed-size representation. The recursive auto-encoder is a recursive neural network that learns feature representations for each node in the tree, such that the word vectors underneath each node can be recursively reconstructed. These feature representations are used to compute a similarity matrix that compares both the single words as well as all non-terminal node features in both sentences. The dynamic pooling layer is used to keep as much of the resulting global information of this comparison as possible, and to deal with the arbitrary length of the two sentences. A softmax classifier, relying on features based on the similarity matrix together with simple features such as the difference in sentence length, or the percentage of words and phrases in one sentence that are in the other sentence and vice-versa, is finally used to classify whether the two sentences are paraphrases or not.

Blacoe and Lapata (2012) also approached the problem of paraphrase identification through the modeling of compositional meanings for sentences, relying on distributional word clustering methods to capture word similarity. The authors experimented with several possible combinations of word representations (e.g., a simple semantic space where a word's vector represents its co-occurrence with neighboring words, a syntax-aware space based on weighted distributional tuples that encode typed co-occurrence relations among words, and word embeddings computed with a neural language model) and composition methods for representing sentences using the vectors of their constituent words (e.g., based on vector addition, multiplication, and recursive neural networks). For each of the three vector sources and three different compositional methods, the authors created the following features: (i) a vector representing the pair of input sentences either via concatenation or subtraction; (ii) a vector encoding which words appear therein; and (iii) a vector made up of the following four other pieces of information: the cosine similarity of the sentence vectors, the length of S_1, the length of S_2, and the unigram overlap among the two sentences. They then used the LibLinear classifier introduced by Fan et al. (2008) to label sentence pairs as either paraphrases or not. The experimental results showed that shallow approaches (i.e., simple semantic space representations) are as good as more computationally intensive alternatives.

The introduction of similarity metrics based on syntactic features derived from dependency parse trees has also been shown to boost performance, under the assumption that if two sentences are paraphrases, their dependency trees should align closely. For instance Wan et al. (2006) employed features based on machine translation metrics (i.e., BLEU-based features) in combination with several other features based on dependency relations and tree edit-distance, inside an SVM classifier. More recently, Das and Smith (2009) used quasi-synchronous dependency grammars to model the structure of the sentences involved in the comparison, and their correspondences. In brief, these authors employ a generative model that generates a paraphrase of a given sentence, and latter use probabilistic inference to reason about whether two sentences share the paraphrase relationship. The model incorporates both syntax and lexical semantics through the formalism of quasi-synchronous dependency grammars, which establish loose links between the syntactic structures of the two sentences, this way allowing for some divergences. Using a product of experts, the authors also combine the proposed model with a complementary logistic regression model, using some of the lexical overlap features that are popular on related works concerning with paraphrases.

Table 1 presents a brief overview on the different approaches that have been surveyed, together with the corresponding evaluation results over the Microsoft Research Paraphrase Corpus (MSRPC), a well-known benchmark dataset in the area, introduced by Dolan et al. (2004).

3 Proposed Method

Our work takes the standard approach of modeling paraphrase identification as a supervised classification problem. We experimented with different learning algorithms based on ensembles of trees (i.e., AdaBoost, Random Forests, and Rotation Forests), for combining a rich set of features that are essentially based on lexical information. The well-known MorphAdorner[1] NLP package was used in the pre-processing of the textual sentences, providing us with English tokenization, stemming, and lemmatization (i.e., a process of reducing inflected spellings to their lexical root that

[1] http://morphadorner.northwestern.edu/

Reference	Brief Description	Algorithm	Acc.	F_1
Mihalcea et al. (2006)	Baseline with cosine similarity and TF-IDF weights	unsupervised	0.654	0.753
Zhang and Patrick (2005)	Lexical features after text canonicalization	supervised	0.703	0.795
Mihalcea et al. (2006)	Combination of word-to-word similarities	unsupervised	0.703	0.813
Rus et al. (2008)	Graph subsumption	unsupervised	0.706	0.805
Qiu et al. (2006)	Sentence dissimilarity classification	supervised	0.720	0.816
Islam and Inkpen (2007)	Combination of semantic and string similarity	unsupervised	0.726	0.813
Blacoe and Lapata (2012)	Semantic spaces from word clustering	supervised	0.730	0.823
Fernando and Stevenson (2008)	Wordnet metric and vector similarity	unsupervised	0.741	0.824
Zia and Wasif (2012)	Semantic heuristic features	supervised	0.747	0.818
Finch et al. (2005)	Combination of MT evaluation measures	supervised	0.750	0.827
Wan et al. (2006)	Dependency-based features	supervised	0.756	0.830
Das and Smith (2009)	Product of experts, using dependency parsing	supervised	0.761	0.827
Kozareva and Montoyo (2006)	Combination of lexical and semantic features	supervised	0.766	0.796
Socher et al. (2011)	Recursive autoencoder with dynamic pooling	supervised	0.768	0.836
Madnani et al. (2012)	Modern machine translation metrics	supervised	0.774	0.841

Table 1: Comparison of different methods that have been previously proposed.

is more advanced than stemming, which for instance reduces different verb forms to the simple infinitive, and which also performs some simple spelling standardization operations).

In terms of the considered learning methods, we choose to experiment with models based on ensembles of trees due to the superior performance shown by these methods on many different learning problems (Rokach, 2010). We used the implementations from the machine learning toolkit named Weka[2] for the considered classification models, and we kept the default parameters defined within this tool for each of the different methods.

For instance the AdaBoost classification algorithm creates the ensemble by generating and calling weak classifiers (i.e., decision stumps) in a series of rounds (Freund and Schapire, 1997). For each call, a distribution of weights over the training examples is updated, indicating the importance of the different examples for the classification. On each round, the weights of each incorrectly classified example are increased, and the weights of each correctly classified example are decreased, so the new classifier focuses on the examples which have so far eluded a correct classification. The final model is a combination of the multiple weak learners, and the classification is given with basis on a weighted majority vote.

Random Forests also operate by constructing a multitude of decision trees at training time, outputting the class that is the mode of the classes output by individual trees (Breiman, 2001). The method combines the ideas of bagging (i.e., choosing a different training set for each tree, by taking n examples with replacement from all N available training cases) and random selection of features (i.e., for each node of each tree, we randomly choose m variables on which to base the decision at that node). Rotation Forests draw upon the same principles behind Random Forests, but each tree is trained on the whole data set, although projected into a rotated feature space (Rodriguez et al., 2006). Bootstrap samples are taken as the training set for the individual classifiers, as in regular bagging. The main heuristic is to apply feature extraction and to subsequently reconstruct a full feature set for each classifier in the ensemble. To do this, the feature set is split randomly into K subsets, and principal component analysis (PCA) is then run separately on each subset. A new set of n linear features is constructed by pooling all principal components, and each classifier in a Rotation Forest is trained with one of these data sets.

As for the considered features, we used different types of similarity metrics involving different representations for the lexical contents of the sentences. We specifically considered representations for the sentences based on:

1. Tokens extracted from lowercased versions of the original sentences;

2. Lemmas extracted from lowercased versions of the original sentences;

3. Stems extracted from lowercased versions of the original sentences;

4. Word clusters associated to the tokens extracted from the sentences, using a total of 100 different clusters;

[2] http://www.cs.waikato.ac.nz/ml/weka/

5. Word clusters associated to the tokens extracted from the sentences, using a total of 320 different clusters;

6. Word clusters associated to the tokens extracted from the sentences, using a total of 3200 different clusters;

7. Double Metaphone phonetic keys (Philips, 2000) associated to the tokens extracted from the original sentences;

8. Character trigrams extracted from the original sentences;

In Items 4, 5 and 6 from the previous enumeration, we considered distributional word clusters obtained through the procedure outlined by Turian et al. (2010), with basis on a large corpus of journalistic texts[3]. When generating our representations for the sentences based on word clusters, we first attempted to use clusters associated to the exact terms that were present in the sentence and, in the case of tokens that were not present in the corpus of journalistic texts, we also attempted to obtain the cluster corresponding to either the lemma or the stem associated to the corresponding token.

Using the eight different representations that were outlined above, we computed features with basis on (i) the cosine similarity between feature vectors, considering TF-IDF weights for each feature, and (ii) the Jaccard similarity coefficient between the sets of features. In order to deal with small variations in terms of word spellings, and besides using the Double Metaphone algorithm, we also computed a set of features based on the Soft-TF-IDF similarity metric, which measures similarity between vector-based representations of the sentences, as in the case of the cosine similarity metric, but considering an internal similarity metric for finding equivalent words. Our features based on Soft-TF-IDF were computed with basis on representations 1,2 and 3, using the Jaro-Winkler similarity metric between words with a threshold of 0.9, as the internal similarity metric.

We also computed features based on the size of the original sentences (i.e., two features, each one encoding the size of each sentence), and the size of the longest common subsequences between the sentences being compared. When computing the longest common subsequences, we consider sentences corresponding to the representations given by all but the last item in the previous enumeration of the representations (i.e., new sentences obtained from the original ones, after replacing the tokens by (i) their lowercased versions, (ii) the corresponding lemmas, (iii) stems, (iv, v and vi) word clusters, and (vi) Double Metaphone phonetic keys).

Leveraging all but the last representation from the enumeration above, we also computed the normalized compression distance as a similarity metric between the sentences (i.e., 7 different features). The normalized compression distance is based on developments related to Kolmogorov complexity, and the basic idea is that if you have two strings, x and y, with some overlap between them, then the concatenated and compressed strings $x + y$ should be smaller than the concatenation of separately compressed strings x and y (Li et al., 2004).

Finally, we also computed features based on common metrics in the area of machine translation, namely (i) BLEU, (ii) METEOR, (iii) WER and (iv) TER. Features corresponding to these four metrics were again obtained from the representations of the sentences corresponding to all but the last item in the previous enumeration.

In brief, we have that BLEU is the most commonly used metric for MT evaluation (Papineni et al., 2002). It is computed as the amount of n-gram overlap, for different values of n (i.e., in our case we used BLEU-3, BLEU-6 and BLEU-9), between the two sentences, tempered by a length penalty. METEOR is a variation of BLEU which uses a combination of both precision and recall, unlike BLEU which focuses on precision (Lavie and Banerjee, 2005). As for the Word Error Rate (WER), it is a simple metric based on dynamic programming that is defined as the number of edits needed to transform one string into the other. The metric known as the Translation Error Rate (TER) differs from WER in that it includes a heuristic algorithm to deal with shifts in addition to insertions, deletions and substitutions (Snover et al., 2006). Although some of these MT evaluation metrics can directly consider the usage of external lexical resources (e.g., WordNet synonyms), we used the simpler versions of the metrics, and computed them over different representations of the original strings (e.g., using lowercased to-

[3] http://metaoptimize.com/projects/wordreprs/

kens, lemmas, stems, word clusters, and Double Metaphone phonetic keys).

In total, our models considered 79 different features representing each sentence pair.

4 Experimental Results

In our experiments, we used the Microsoft Research Paraphrase Corpus (MSRPC) introduced by Dolan et al. (2004). The corpus consists of a large set of sentence pairs $\{S_1, S_2\}$, together with labels indicating whether the sentences are in a paraphrase relationship or not. The MSRPC dataset contains 5,801 sentence pairs, and we used the standard split of 4,076 training pairs (67.5% of which are paraphrases) and 1,725 test pairs (66.5% of which are paraphrases).

Brockett and Dolan (2005) remark that this corpus was created semi-automatically by first training an SVM classifier on a disjoint annotated dataset containing 10,000 sentence pairs, and then applying the SVM on an unseen 49,375 sentence pair corpus, with its output probabilities skewed towards over-identification, i.e., towards generating some false paraphrases. A total of 5,801 sentence pairs out of these 49,375 pairs were randomly selected and presented to human judges for refinement into true and false paraphrases.

A strict definition of a paraphrase would insist on the two candidate texts having identical meanings. However the creators of the MSRPC found that this strict definition would limit paraphrase pairs to be virtually identical string copies of each other. Such pairs provide interesting data for analyzing minor syntactic and lexical alternations, but the authors of the MSRPC wanted to capture more interesting and complex differences. This led them to relaxing the definition of a paraphrase from *full bidirectional entailment* to *mostly bidirectional entailments*, and a total of 3,900 pairs were marked by the human judges as having *mostly bidirectional entailments*. Still, a key message of the guidelines for the annotators of the corpus stated that in order to constitute a paraphrase, sentence pairs should describe the same event and contain the same important information about that event. Each sentence was labeled first by two judges, who averaged 83% agreement, and a third judge resolved conflicts. This average agreement can be considered as an upper bound for the accuracy that can be obtained using automatic methods for performing the identification.

Algorithm	Accuracy	F_1
AdaBoost	0.704	0.767
Random Forest	0.748	0.825
Rotation Forest	0.773	0.837

Table 2: Results with different learning methods.

Strategy	Accuracy	F_1	# Features
Forward	0.752	0.832	8
Backward	0.773	0.837	79

Table 3: Results with greedy feature selection.

In a first set of experiments, we analyzed the performance obtained with different learning algorithms and the full set of features described in Section 3. Table 2 presents the obtained results, showing that a Rotation Forest classifier obtained slightly better results.

In a second set of experiments, we tried to analyze the relative merits of the different features, through greedy-forward and greedy-backward feature selection strategies, together with a classification model based on Rotation Forests. Greedy-forward feature selection adds one feature dimension at a time to a set of already selected features, and checks how good that feature is by training and testing a classifier. The best feature is then added to the set of selected features, and the process stops when adding another feature does not result in a better classifier. Greedy-backward feature elimination is a similar procedure, in which we instead start with the full set of features and, one-by-one, remove the features that least contribute to classification accuracy, until the quality of the results drops bellow an accepted threshold. Table 3 shows the obtained results, presenting the number of features that was kept by each strategy, together with the obtained results in terms of the Accuracy and F_1 metrics.

The 8 features selected by the Forward strategy include the TF/IDF and Soft-TF/IDF scores computed with the lowercased tokens, the longest common subsequence computed with basis on word clusters, and different machine translation metrics computed with basis on lemmas, stems, and word clusters. The most discriminative feature was found to be the TF/IDF score.

From the selected features we can conclude that (i) the sentences had few spelling variations and the added value of using the Double Metaphone

keys was low, (ii) five out of eight features selected by the greedy-forward procedure were machine translation metrics, (iii) seven out of eight features are values that come from different types of metrics, and (iv) TF-IDF (or Soft TF-IDF) are the best individual algorithms to analyze whether two sentences are paraphrases of each other.

We also manually analyzed some of the sources of error in the identification of paraphrases. Table 4 presents some of the sentence pairs that were incorrectly classified. The first two pairs show that there are sentences in the corpus with very similar words and that apparently refer to the same entities and events, but that are nonetheless marked as not being paraphrases in the MSRPC. The last two examples show that issues such as grammatical voice or the usage of a different vocabulary constitute important sources of error.

Besides experiments with the Microsoft Research Paraphrase Corpus, we have also made some experiments with data from a practical application domain, where paraphrase identification is an important issue, namely with the mono-lingual English dataset from the FireFAQ shared task on FAQ retrieval (Contractor et al., 2013). The goal of this task is to find a question Q from a corpus of FAQs that best answers/matches a noisy question S (i.e., the task considers queries which are written in *SMS language*, where users try to compress text by omitting letters, using slang, etc.). We therefore model the task in the same way we did for paraphrase identification, using a classifier to combine multiple similarity metrics between the SMS questions S and the questions Q in the corpus of FAQs. We return up to five questions that are labeled as paraphrases, ordered according to the confidence score returned by the classifier.

In order to deal with SMS-specific slang for the FireFAQ task, we used a step of SMS language normalization, in which a list of 2561 SMS abbreviations was used to replace slang words with their regular English equivalents (e.g., "btw" is replaced for "by the way"). SMS messages can also frequently contain unintended typographical errors (e.g., due to small size of the keypads on mobile phones), and we try to address this issue through the similarity metrics that rely on the Double Metaphone phonetic matching algorithm, which were already considered in the experiments with the MSRPC dataset. We used the complete set of features from Section 3, as well as the representations that were described in that section, after an initial pre-processing set in which we replaced the SMS language using the list of abbreviations.

	P@1	P@5	MRR
FireFAQ 2011	0.879	0.899	0.981
FireFAQ 2012	0.725	0.818	0.943

Table 5: Results on the FireFAQ data.

Table 5 presents the obtained results on the datasets from the FireFAQ 2011 and 2012 competitions, when considering a model based on a Rotation Forest classifier, and when considering the full set of features introduced in Section 3. The evaluation results are presented in terms of three metrics, namely the average precision at cut-off positions 1 and 5 (i.e., seeing if the correct FAQ question is returned as the top result, or in one of the top five results), and the Mean Reciprocal Rank (MRR) for the correct result, in the top 5 questions that are returned by the system. In the FireFAQ datasets, some of the considered queries had a corresponding question in the database of FAQs, while others had no such correspondents (i.e., in FireFAQ 2011, only 728 queries, in a total of 3405 queries, had a corresponding question in the database of FAQs). Our MRR evaluation metric was computed with basis only on the queries containing a true corresponding question in the database. Still, in the case of the P@1 and P@5 metrics, the cases where the queries had no correspondent, and in which our system also did not return any paraphrase, were considered as correct results.

The results in Table 5 show that our approach based on paraphrase detection is able to achieve good results on the FAQ retrieval task.

In FireFAQ 2011, the best participating system achieved an MRR score of 0.896, a slightly inferior result in comparison with ours. The median MRR score was of 0.14. The best participating system correctly identified the answer to 494 queries having a corresponding answer in the database, and it correctly identified 2311 queries as not having a corresponding answer, whereas our method obtained the correct answer to 509 queries, and it correctly identified 2562 queries that were not answered in the database of FAQs.

Sentence1	Sentence2	is Paraphrase	Evaluation
Crews worked to install a new culvert and prepare the highway so motorists could use the eastbound lanes for travel as storm clouds threatened to dump more rain.	Crews worked to install a new culvert and repave the highway so motorists could use the eastbound lanes for travel.	false	true
Bethany Hamilton remained in stable condition Saturday after the attack Friday morning.	Bethany, who remained in stable condition after the attack Friday morning, talked of the attack Saturday.	false	true
Remaining shares will be held by QVC's management.	Members of the QVC management team hold the remaining shares.	true	false
Mr. Malik assured him that he would be considered a martyr if he did not return, the witness testified.	Mr. Malik assured him that he would be considered a martyr if anything happened to him as a result of his trip, the witness said.	true	false

Table 4: Examples of incorrectly evaluated sentences.

5 Conclusions and Future Work

We proposed a paraphrase identification method based on supervised machine learning, which involves training a classifier that uses features corresponding to string similarity metrics that mostly rely on lexical information. The most innovative contributions behind our work relate to (i) the usage of state-of-the-art classification methods based on tree ensembles, (ii) the combination of machine translation metrics with other types of features, and (iii) the usage of string similarity features based on distributional word clustering. We report on a set of experiments that used the well-known Microsoft Research Paraphrase Corpus, in which we achieved a classification accuracy of 0.77, and an F1 measure of 0.84. We have therefore shown that out-of-the-box learning algorithms and relatively simple features can obtain state-of-the-art results in this task.

Despite the interesting results, there are also several ideas for future improvements. We would, for instance, like to experiment with the usage of rules to transform sentences using different grammatical patterns (i.e., transforming passive voice into active voice), as well as with the introduction of other features based on evaluation metrics from the areas of machine translation, text summarization, or sequence alignment in general.

For future work, we would also like to perform experiments on different datasets (e.g., on texts from other languages) and on different domains that also involve the computation of semantic relatedness between sentences. One of these domains concerns, for instance, with the identification of metaphors, as shown in the previous studies by Shutova et al. (2012) and by Bollegala and Shutova (2013).

References

W. Blacoe and M. Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and on Computational Natural Language Learning*, 2012.

D. Bollegala and E. Shutova. Metaphor Interpretation Using Paraphrases Extracted from the Web. *PLoS ONE*, 8(9), 2013.

L. Breiman. Random forests. *Machine Learning*, 45(1), 2001.

C. Brockett and W. B. Dolan. Support Vector Machines for Paraphrase Identification and Corpus Construction. In *Proceedings of the 3rd International Workshop on Paraphrasing*, 2005.

D. Contractor, L. V. Subramaniam, P. Deepak, and A. Mittal. Text Retrieval Using SMS Queries: Datasets and Overview of FIRE 2011 Track on SMS-Based FAQ Retrieval. In *Proceedings of the Forum for Information Retrieval Evaluation*, 2013.

D. Das and N. A. Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings*

of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and of the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, 2009.

B. Dolan, C. Quirk, and C. Brockett. Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics*, 2004.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal Machine Learning Research*, 9, 2008.

S. Fernando and M. Stevenson. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium on Computational Linguistics in the UK*, 2008.

A. Finch, Y. H, and E. Sumita. Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In *Proceedings of the International Workshop on Paraphrasing*, 2005.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 1997.

A. Islam and D. Inkpen. Second order co-occurrence PMI for determining the semantic similarity of words. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2006.

A. Islam and D. Inkpen. Semantic similarity of short texts. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, 2007.

Z. Kozareva and A. Montoyo. Paraphrase identification on the basis of supervised machine learning techniques. In *Proceedings of the 5th International Conference on Advances in Natural Language Processing*, 2006.

A. Lavie and S. Banerjee. Meteor: an automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the 2nd Workshop on Statistical Machine Translation*, 2005.

M. Li, X. Chen, X. Li, B. Ma, and P. M. B. Vitanyi. The similarity metric. *IEEE Transactions on Information Theory*, 50(12), 2004.

N. Madnani, J. Tetreault, and M. Chodorow. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2012.

R. Mihalcea, C. Corley, and C. Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.

L. Philips. The double metaphone search algorithm. *C/C++ Users Journal*, 18(6), 2000.

L. Qiu, M.-Y. Kan, and T.-S. Chua. Paraphrase recognition via dissimilarity significance classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2006.

J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso. Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (10), 2006.

L. Rokach. Ensemble-based classifiers. *Artificial Inteligence Review*, 33(1-2), 2010.

V. Rus, P. McCarthy, M. Lintean, D. McNamara, and A. Graesser. Paraphrase identification with lexico-syntactic graph subsumption. In *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*, 2008.

E. Shutova, T. Van de Cruys, and A. Korhonen. Unsupervised Metaphor Paraphrasing using a Vector Space Model. In *Proceedings of International Conference on Computational Linguistics*, 2012.

M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, 2006.

R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proceedings of the Conference on Neural Information Processing Systems*, 2011.

J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

S. Wan, M. Dras, R. Dale, and C. Paris. Using dependency-based features to take the 'para-farce' out of paraphrase. In *Proceedings of the 4th Australasian Language Technology Workshop*, 2006.

Y. Zhang and J. Patrick. Paraphrase identification by text canonicalization. In *Proceedings of the 3rd Australasian Language Technology Workshop*, 2005.

U.-Q. Zia and A. Wasif. Paraphrase Identification using Semantic Heuristic Features. *Research Journal of Applied Sciences, Engineering and Technology*, 4(22), 2012.