Paraphrase Identification on the Basis of Supervised Machine Learning Techniques

Zornitsa Kozareva and Andrés Montoyo

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante
{zkozareva, montoyo}@dlsi.ua.es

Abstract. This paper presents a machine learning approach for paraphrase identification which uses lexical and semantic similarity information. In the experimental studies, we examine the limitations of the designed attributes and the behavior of three machine learning classifiers. With the objective to increase the final performance of the system, we scrutinize the influence of the combination of lexical and semantic information, as well as techniques for classifier combination.

1 Introduction and Related Work

Natural language is the most powerful tool through which people establish communication and relate to each other. In our daily life we can use different words and phrases to express the same meaning. This is related to our knowledge and cultural habits, that later reflect on our written and spoken skills.

The web is the largest text repository, where millions of people share and consult information daily. In the context of Information Retrieval, given a natural language query, the search engine should identify and return documents that have similar or related meanings to the query. However, the relevant information may be present in different forms. For example a search about "operating systems" should retrieve document about "unix". In order to identify that although neither "operating" nor "systems" appear, the document is still relevant as "unix" is a type of operating system, a paraphrase identification module is needed.

Other Natural Language Processing (NLP) applications such as Information Extraction (IE) or Question Answering (QA) also have to handle lexical, semantic or syntactic variabilities. Thus, they avoid the usage of redundant information during the template filling process or find easily the correct answer which may be presented in an indirect way. Experimental studies [12] demonstrate that the identification of language variabilities is important for many NLP areas and their resolution improves the performance of the systems.

Recent paraphrase identification approaches [2] use multiple translations of a single language, where the source language guarantees the semantic equivalence in the target language. In order to extract paraphrases, [20] used named entity anchors, while [1] employed Multiple Sequence Alignment. [11] mined the web

T. Salakoski et al. (Eds.): FinTAL 2006, LNAI 4139, pp. 524–533, 2006.

to obtain verb paraphrases, while [10] constructed a broad-domain corpus of aligned paraphrase pairs through the web. [15] presented a lightweight method for unsupervised paraphrase extraction from billions of web documents.

In this paper, we focus on the paraphrase identification rather than on the paraphrase generation task. Our task consists in given two text fragments, the system has to determine weather the two texts paraphrase each other or not. For example the sentences "James sells four papers to Post International" and "Post International receives papers by James" express the same meaning therefore, they are paraphrases of each other.

Our approach is similar this of [3] who use an annotated dataset and Support Vector Machines to induce larger monolingual paraphrase corpus from a comparable corpus of news clusters found on the web. We rely on already compiled paraphrase corpus [18], so our task reduces to the identification of sentences that are paraphrases of each other, for example "the glass is half-empty" and "the glass is half-full". For this purpose, we develop a supervise machine learning approach where three classifiers are employed. The classifiers use lexical and semantic similarity information. In comparison to [6] who recognize paraphrases measuring text semantic similarity, we capture word semantic similarity.

The novelty of our approach consists in the performed experiments. First we explore the discriminating power of the individual lexical and semantic feature sets to identify paraphrases. In addition, we study the behavior of the three different machine learning classifiers with the modelled features. With the objective to improve the performance of the paraphrase identification system, we examine the impact of the combination of the lexical and semantic surface information in a big feature set and also through voting. Previous researchers did not study the effect of such combinations, therefore we believe that the direction of our approach is novel.

The paper is organized in the following way. Section 2 describes the paraphrase identification system. Section 3 outlines the paraphrasing data we worked with. The next section concerns the conducted experimental setups and finally the conclusions are exposed in Section 5.

2 The Paraphrasing System at a Glance

Most systems [9] used numerous thresholds to decide definitely whether two sentences are similar and infer the same meaning. This threshold determination process is dependent on the training data and apart may lead to incorrect paraphrase reasoning. In order to avoid the threshold settings, we use machine learning techniques. The advantages of a ML approach consists in the ability to account for a large mass of information and the possibility to incorporate different information sources such as morphologic, syntactic, semantic among others in one single execution. The major obstacle for the usage of ML techniques concerns the availability of training data. For our approach we used a standard paraphrase evaluation corpus therefore, learning from the data examples was possible.

Thus, it was reasonable to propose and possible to develop a machine learning based paraphrase identification approach. Figure 1 shows the modules of the paraphrase system.

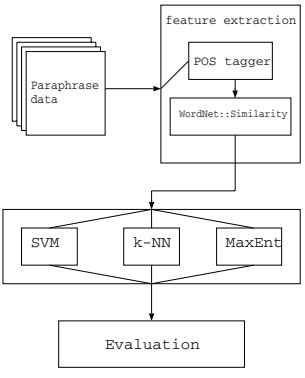

Fig. 1. Modules of the paraphrase identification system

2.1 Feature Extraction

The most important module in a machine learning system concerns the feature extraction and generation one. To perform well, every machine learning classifier needs relevant attributes calculated from the instances in the data set. For this reason, we start the description of our paraphrasing system from the feature extraction module.

As paraphrases appear on lexical, syntactic, semantic and pragmatic levels, or in a combination among them, we explore the discriminating power which can be obtained on the lexical and semantic similarity levels. All of the designed attributes capture the sentence similarity in both directions, because paraphrases are bidirectional relations [10].

The *word overlap feature set* includes well known text summarization measures. The first two attributes establish the ratio of the common consecutive *n-grams* between two texts T_1 and T_2[1], against the total number of words in T_1/T_2. For this feature, the high number of common words indicates that the two sentences are similar and we interpret it as high probability for the two sentences to paraphrase each other. However, unigrams alone fail to identify that "Mary calls the police" and "the police calls Mary" do not infer the same meaning. Therefore, to identity better the proximity of the sentences, we employ attributes sensitive to word order. Two such measures we found are the skip-gram and the longest common subsequence.

Skip-grams look for non consecutive sequences of words that may have gaps in between, compared to all combinations of words that can appear in the

[1] T_1 refers to the first sentence and T_2 refers to the second sentence.

sentences. The two measures are $skip\_gramT_1 = \frac{skip\_gram(T_1,T_2)}{C(n,skip\_gram(T_1,T_2)}$ and $skip\_gramT_2 = \frac{skip\_gram(T_1,T_2)}{C(m,skip\_gram(T_1,T_2)}$. The $skip\_gram(T_1, T_2)$ refers to the number of common skip grams (pair of words in sentence order that allow arbitrary gaps) found in T_1 and T_2 and $C(n, skip\_gram(T_1, T_2))$ is a combinatorial function, where n is the number of words in text T_1 (e.g. m corresponds to the number of words in T_2). The maximum length of the skip-gram calculation is restricted to four, because sequences higher than this do not appear very often. This measure is known in text summarization as ROUGE-S [13].

The *longest common subsequence* (LCS) determines one[2] long common subsequence of words between two sentences. Once the LCS is found, it is normalized by the number of words present in T_1/T_2. The ratio indicates how many non consecutive words appear between the two sentences in respect to all words.

So far, the presented surface features are designed to capture lexical variations. As counting n-grams is not a language dependent task, this allows their application to the recognition of paraphrases or text entailments [8] for other languages.

In order to obtain the semantic similarity attributes, first we determined the parts-of-speech tags with the TreeTagger [19] toolkit. *Word similarity features* need extrinsic knowledge which can be collected from large corpora or word repository as WordNet[3]. To establish the similarity among the nouns and verbs in the sentences, we used the WordNet::Similarity package [16] with the measure of [14].

We introduce a noun/verb semantic similarity measure obtained with the calculation of the formula $sim_{lin} = \frac{\sum_{i=1}^{n} sim(T_1,T_2)_{lin}}{n}$. This measure indicates the ratio of the noun/verb similarity with respect to the maximum noun/verb similarity for the sentences T_1 and T_2. The values of $sim(T_1, T_2)_{lin}$ are the similarity of noun/verb pairs for the text T_1 and T_2 according to the measure of [14]. For perfect similarity match, sim_{lin} has value 1 and for completely dissimilar words 0.

The *cardinal number* attribute captures that "more than 24" indicates 25 and the numbers above it, "less than 24" is 23 and the numbers below it. Writing as "twenty-five" is transformed automatically into "25", and then is lexically matched with the corresponding number. When the texts contain several cardinal numbers, this attribute matches from all possible numbers how many coincidences the two texts have.

The *proper name* attribute is 1 for perfect proper name matches such as "London" and "London", and 0 for sentences where there are no proper names at all, or when the proper names are completely distinct.

When the described features are generated for each paraphrase pair in the MSP corpus, the functioning of the feature module is terminated and the machine learning module is initiated. In the next subsection, we describe the classifiers used for the training and testing phases.

[2] If LCS finds two different longest common subsequence strings of the same length, only one of them is taken.

[3] wordnet.princeton.edu/

2.2 Machine Learning Module

A machine learning module can be composed of different number of classifiers. For our system, we selected three algorithms based on their processing time and generalization function.

Support Vector Machines (SVM) are known to perform well with two class problems, with high data sparcity and multiple attribute space. As parphrase recognition reduces to a two class problem, we decide that the utilization of SVM is pertinent. The software we worked with is called SVMTorch [5]. Several kernels were tested and the best performing one was the linear.

k-Nearest Neighbors (k-NN) is a lazy learner that stores every training example in the memory. This algorithm is useful when the number of training examples is not sufficient. During testing, a new case is classified by extrapolating the most similar stored examples. The similarity between a new instance X and all examples Y in the memory is computed by the distance metric $\triangle(X,Y) = \sum_{i=1}^{n} \delta(x_i, y_i)$, where $\delta(x_i, y_i) = |\frac{x_i - y_i}{max_i - min_i}|$. We used the Memory-based learning algorithm developed by [7].

Maximum Entropy (MaxEnt) estimates probabilities based on the principle of making as few assumptions as possible. The probability distribution that satisfies the above property is the one with the highest entropy. An advantage of MaxEnt framework is that even knowledge-poor features can be applied accurately. We used the MaxEnt implementation of [21].

3 Data Set and Evaluation

We evaluate the performance of our machine learning paraphrase identification system on a standard paraphrase corpus developed and provided by Microsoft[4] [18].

This corpus consists of training and testing data sets. Each line has two sentences, and the paraphrase identification task consists in determining whether these two sentences are paraphrases of each other or not. The training set consists of 4076 sentence pairs, of which 2753 are paraphrases of each other. The testing set has 1726 sentence pairs, of which 1147 are paraphrases of each other.

The evaluation measures are the traditional precision, recall and f-score. Systems are ranked and compared according to the accuracy score, which indicates the number of correct responses in respect to all test entries.

4 Experiments

Three types of experiments were conducted to answer the questions: Which machine learning algorithm is the most reliable with the presented feature sets? Does the mixture of lexical and semantic information lead to improvement? What happens through multiple classifier combination?

[4] http://research.microsoft.com/research/downloads/

4.1 Experimental Setup 1

As previously mentioned, to construct a robust multilevel paraphrase system, the resolution power of the individual machine learing classifiers should be explored. In our first experiment, we study the performance of the three machine learning algorithms with the designed *word overlap* and *word similarity* feature sets.

Initially, the three classifiers SVM, k-NN and MaxEnt were trained and tested with the *word overlap* feature set. The obtained results for the whole paraphrase identification test corpus are shown in Table 1.

Table 1. Paraphrase identification with word overlap information

System	Acc.	Prec.	Rec.	F-score
SVM	69.86	93.46	70.66	80.48
MaxEnt	68.29	69.16	59.53	63.98
k-NN	63.36	74.45	71.58	72.99
C-M	68.80	74.10	81.70	77.70
word match	66.10	72.20	79.80	75.80

Although the three classifiers use the same attributes, the yielded performances are different due to their varied machine learning philosophy. In our task, we deal with two class problem. For this experiment, the obtained results showed that the word overlap feature set indicated correctly most of the examples that do not paraphrase each other. This is related to the fact that the word overlap features penalize longer sentences as they cannot map the majority of the words.

The best generalization among all classifiers is achieved by SVM. MaxEnt and k-NN algorithms gained 68.29% and 63.36% accuracy. Comparing these results to a baseline that counts the number of common words, only k-NN could not outperform it.

In the same table, we compare the obtained results to the system of [6]. We denote their system as C-M. Although C-M measured text semantic similarity, and in our approach we compute word overlaps, the SVM run achieved better f-score and accuracy coverage. This indicates that the modelled attributes are good indicators for paraphrase identification.

A positive characteristics of the word overlap feature set is that it is simple to implement and has low computational cost. The feature set is language independent, because counting words is not a language dependent task. This property makes it easy and practical to be applied to languages with limited resources. However, a negative aspect of the lexical features is that their performance cannot be improved anymore.

For the *word similarity* feature set, the obtained results are shown in Table 2. According to the accuracy measures, three machine learning classifiers performed worse than the system of [6], but comparing the f-scores SVM performs better than C-M. One reason for the low performance is that only word to word similarity is not informative enough to identify paraphrases. In contrast to the

Table 2. Paraphrase identification with word similarity information

System	Accuracy	Prec.	Rec.	F-score
SVM	66.50	100	66.49	79.87
MaxEnt	66.49	81.15	68.20	74.11
k-NN	67.81	91.30	66.43	76.90
C-M	68.80	74.10	81.70	77.70
word match	66.10	72.20	79.80	75.80

word overlap set that determined correctly most of the non paraphrase pairs, the semantic set identified correctly the sentences that paraphrase each other. This is due to the sim_{lin} measure according to which if there is one completely similar noun/verb pair or most of the noun/verb pairs are similar, then the sentences paraphrase each other.

4.2 Experimental Setup 2

In this experimental setup, we study the combination of the lexical and semantic similarity information into a single feature set. The achieved results are shown in Table 3.

Table 3. Paraphrase identification with the combination of word overlap and similarity features

System	Accuracy	Prec.	Rec.	F-score
SVM	70.43	84.66	74.12	79.04
MaxEnt	66.44	82.13	70.50	75.87
k-NN	64.68	78.88	71.13	74.81
C-M	68.80	74.10	81.70	77.70

Compared to the previous results, in this experiment the classifiers determined correctly equally paraphrasing and non paraphrasing sentences. The best performing classifier is SVM. Only for it, the combination of word overlap and semantic features lead to increase in performace with around 1%. According to z' statistics, such improvement is insignificant. When we saw that the feature combination did not help, we performed another experiment where the generated outputs of the lexical and semantic classifiers are combined through voting.

4.3 Experimental Setup 3

For the voting scheme first the outputs of the generated lexical and semantic SVM, k-NN and MaxEnt classifiers are examined. There, test cases whose classes coincided by the two of the three classifiers, directly obtain the majority class. For the instances where the two classifiers disagree, the class of the classifier with the highest performance was adopted. The obtained results of the voting executions are shown in Table 4.

Table 4. Paraphrase identification with voting

System	Accuracy	Prec.	Rec.	F-score
SVM,k-NN,MaxEnt	76.64	94.42	68.76	79.57
C-M	68.80	74.10	81.70	77.70

According to the statistical z' test [5], the classifiers' accuracy significantly improved with voting. This improvement is due to the high complementarity of the lexical and semantic feature sets, which according to the kappa statistical measure [4] complement each other. Similar approach for complementarity examination was used by [17] who determined how to combine different word sense disambiguation systems in a beneficial way.

Through the experimental setups, we show word overlaps can identify correctly sentences that do not paraphrase each other. In addition, the combination of the lexical and semantic attributes in a single feature set did not enrich the performance. However, the combination of the lexical and semantic information through voting was beneficial. Finally, in a comparative study, we demonstrate that the proposed machine learning paraphrase identification approach can outperform more complex method like [6] which tries to measure text semantic similarity.

5 Conclusion and Future Work

We presented a machine-learning approach for the paraphrase identification task. Three machine learning algorithms were used to determine which one of them is the most appropriate for the paraphrase task. Several experiments were conducted and the obtained results were compared to a baseline and already existing systems.

The experiments revealed that simple features relying on common consecutive or insequence matches can resolve correctly 69.86% of the paraphrases. Such attributes are very useful and practical for languages with scarce resources. Unfortunately, on their own these attributes cannot be improved any more.

The combination of lexical and semantic attributes into a single feature set did not improve the accuracy of the different machine learning classifiers. Therefore, we studied a better way to combine this information. The used voting algorithm that boosted the final performance with 10%. According to z' statistics, this improvement is significant compared to the single classifier.

For all experiment, the best performance is obtained with SVM. We consider its usage for the paraphrase identification as very proper. With the analysis of the results, we saw that this is due to the ability of SVM to work with high dimensional attribute spaces.

In the future, we want to incorporate a Named Entity Recognizer which will improve the performance of the proper name attribute. As paraphrases act on different representation levels – lexical, semantic, syntactic or even a combination

[5] The tested confidence was 98%.

among them all, we believe that the incorporation of syntactic information is going to be helpful for the proposed and developed approach.

Acknowledgements

This research has been partially funded by the Spanish Government under project CICyT number TIC2003-07158-C04-01 and PROFIT number FIT-340100-2004-14 and by the Valencia Government under project numbers GV04B-276.

References

1. Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL 2003: Main Proceedings*, pages 16–23, 2003.
2. Regina Barzilay and Kathleen McKeown. Extracting paraphrases from a parallel corpus. In *39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57, 2001.
3. Chris Brockett and William B. Dolan. Support vector machines for paraphrase identification and corpus construction. In *Second International Joint Conference on Natural Language Processing*.
4. Jacob Cohen. A coefficient of agreement for nominal scales. *Educ. Psychol. Meas*, 1960.
5. Ronan Collobert and Samy Bengio. Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 1, issn 1533-7928:143–160, 2001.
6. Courtney Corley and Rada Mihalcea. Measures of text semantic similarity. In *Proceedings of the ACL workshop on Empirical Modeling of Semantic Equivalence*.
7. Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. Timbl: Tilburg memory-based learner. Technical Report ILK 03-10, Tilburg University, November 2003.
8. Ido Dagan and Oren Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. In *PASCAL Workshop on Learning Methods for Text Understanding and Mining*.
9. Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.
10. William B. Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *International Conference on Computational Linguistics, COLING*.
11. Oren Glickman and Ido Dagan. Acquiring lexical paraphrases from a single corpus. In *Recent Advances in Natural Language Processing III*.
12. Zornitsa Kozareva and Andrés Montoyo. The role and resolution of textual entailment in natural language processing applications. In *11th International Conference on Applications of Natural Language to Information Systems (NLDB 2006)*, 2006.
13. Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, 2003.

14. Dekang Lin. An information-theoretic definition of similarity. In *Proceddings of 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.
15. Marius Pasca and Péter Dienes. Aligning needles in a haystack: Paraphrase acquisition across the web. In *IJCNLP*, pages 119–130, 2005.
16. Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*.
17. Ted Pedersen. Assessing system agreement and instance difficulty in the lexical sample tasks of senseval-2. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
18. Chris Quirk, Chris Brockett, and William B. Dolan. Monolingual machine translation for paraphrase generation,. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
19. Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994.
20. Yusuke Shinyama, Satoshi Sekine, Kiyoshi. Sudo, and Ralf Grishman. Automatic paraphrase acquisition from news articles. 2002.
21. Armando Suárez and Manuel Palomar. A maximum entropy-based word sense disambiguation system. In *COLING*, 2002.