A Generate and Rank Approach to Sentence Paraphrasing

Prodromos Malakasiotis[*] and **Ion Androutsopoulos**[*+]
[*]Department of Informatics, Athens University of Economics and Business, Greece
[+]Digital Curation Unit – IMIS, Research Centre “Athena”, Greece

Abstract

We present a method that paraphrases a given sentence by first generating candidate paraphrases and then ranking (or classifying) them. The candidates are generated by applying existing paraphrasing rules extracted from parallel corpora. The ranking component considers not only the overall quality of the rules that produced each candidate, but also the extent to which they preserve grammaticality and meaning in the particular context of the input sentence, as well as the degree to which the candidate differs from the input. We experimented with both a Maximum Entropy classifier and an SVR ranker. Experimental results show that incorporating features from an existing paraphrase recognizer in the ranking component improves performance, and that our overall method compares well against a state of the art paraphrase generator, when paraphrasing rules apply to the input sentences. We also propose a new methodology to evaluate the ranking components of generate-and-rank paraphrase generators, which evaluates them across different combinations of weights for grammaticality, meaning preservation, and diversity. The paper is accompanied by a paraphrasing dataset we constructed for evaluations of this kind.

1 Introduction

In recent years, significant effort has been devoted to research on paraphrasing (Androutsopoulos and Malakasiotis, 2010; Madnani and Dorr, 2010). The methods that have been proposed can be roughly classified into three categories: (i) *recognition* methods, i.e., methods that detect whether or not two input sentences or other texts are paraphrases; (ii) *generation* methods, where the aim is to produce paraphrases of a given input sentence; and (iii) *extraction* methods, which aim to extract paraphrasing rules (e.g., “X wrote Y” “$\leftrightarrow$ Y was authored by X”) or similar patterns from corpora. Most of the methods that have been proposed belong in the first category, possibly because of the thrust provided by related research on textual entailment recognition (Dagan et al., 2009), where the goal is to decide whether or not the information of a given text is entailed by that of another. Significant progress has also been made in paraphrase extraction, where most recent methods produce large numbers of paraphrasing rules from multilingual parallel corpora (Bannard and Callison-Burch, 2005; Callison-Burch, 2008; Zhao et al., 2008; Zhao et al., 2009a; Zhao et al., 2009b; Kok and Brockett, 2010). In this paper, we are concerned with paraphrase generation, which has received less attention than the other two categories.

There are currently two main approaches to paraphrase generation. The first one treats paraphrase generation as a machine translation problem, with the peculiarity that the target language is the same as the source one. To bypass the lack of large *monolingual* parallel corpora, which are needed to train statistical machine translation (SMT) systems for paraphrasing, monolingual clusters of news articles referring to the same event (Quirk et al., 2004) or other similar monolingual comparable corpora can be used, though sentence alignment methods for parallel corpora may perform poorly on comparable corpora (Nelken and Shieber, 2006); alternatively, large collections of paraphrasing rules obtained via paraphrase extraction from multilingual parallel corpora can be used as monolingual phrase tables in a

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 96–106,
Edinburgh, Scotland, UK, July 27–31, 2011.

phrase-based SMT systems (Zhao et al., 2008; Zhao et al., 2009a); in both cases, paraphrases can then be generated by invoking an SMT system's decoder (Koehn, 2009). A second paraphrase generation approach is to treat existing machine translation engines as black boxes, and translate each input sentence to a pivot language and then back to the original language (Duboue and Chu-Carroll, 2006). An extension of this approach uses multiple translation engines and pivot languages (Zhao et al., 2010).

In this paper, we investigate a different paraphrase generation approach, which does not produce paraphrases by invoking machine translation system(s). We use an existing collection of monolingual paraphrasing rules extracted from multilingual parallel corpora (Zhao et al., 2009b); each rule is accompanied by one or more scores, intended to indicate the rule's *overall* quality without considering particular contexts where the rule may be applied. Instead of using the rules as a monolingual phrase table and invoking an SMT system's decoder, we follow a generate and rank approach, which is increasingly common in several language processing tasks.[1] Given an input sentence, we use the paraphrasing rules to generate a large number of candidate paraphrases. The candidates are then represented as feature vectors, and a ranker (or classifier) selects the best ones; we experimented with a Maximum Entropy classifier and a Support Vector Regression (SVR) ranker.

The vector of each candidate paraphrase includes features indicating the overall quality of the rules that produced the candidate, the extent to which the rules preserve grammaticality and meaning in the particular context of the input sentence, and the degree to which the candidate's surface form differs from that of the input; we call the latter factor *diversity*. The intuition is that a good paraphrase is grammatical, preserves the meaning of the original sentence, while also being as different as possible.

Experimental results show that including in the ranking (or classification) component features from an existing paraphrase recognizer leads to improved results. We also propose a new methodology to evaluate the ranking components of generate-and-rank paraphrase generators, which evaluates them across different combinations of weights for grammaticality, meaning preservation, and diversity. The paper is accompanied by a new publicly available paraphrasing dataset we constructed for evaluations of this kind. Further experiments indicate that when paraphrasing rules apply to the input sentences, our paraphrasing method is competitive to a state of the art paraphrase generator that uses multiple translation engines and pivot languages (Zhao et al., 2010).

We note that paraphrase generation is useful in several language processing tasks. In question answering, for example, paraphrase generators can be used to paraphrase the user's queries (Duboue and Chu-Carroll, 2006; Riezler and Liu, 2010); and in machine translation, paraphrase generation can help improve the translations (Callison-Burch et al., 2006; Marton et al., 2009; Mirkin et al., 2009; Madnani et al., 2007), or it can be used when evaluating machine translation systems (Lepage and Denoual, 2005; Zhou et al., 2006; Kauchak and Barzilay, 2006; Padó et al., 2009).

The remainder of this paper is structured as follows: Section 2 explains how our method generates candidate paraphrases; Section 3 introduces the dataset we constructed, which is also used in subsequent sections; Section 4 discusses how candidate paraphrases are ranked; Section 5 compares our overall method to a state of the art paraphrase generator; and Section 6 concludes.

2 Generating candidate paraphrases

We use the approximately one million English paraphrasing rules of Zhao et al. (2009b). Roughly speaking, the rules were extracted from a parallel English-Chinese corpus, based on the assumption that two English phrases e_1 and e_2 that are often aligned to the same Chinese phrase c are likely to be paraphrases and, hence, they can be treated as a paraphrasing rule $e_1 \leftrightarrow e_2$.[2] Zhao et al.'s method actually operates on slotted English phrases, obtained from parse trees, where slots correspond to part of speech (POS) tags. Hence, rules like the following three may be obtained, where NN_i indicates a noun slot and NNP_i a proper name slot.

[1] See, for example, Collins and Koo (2005).

[2] This pivot-based paraphrase extraction approach was first proposed by Bannard and Callison-Burch (2005). It underlies several other paraphrase extraction methods (Riezler et al., 2007; Callison-Burch, 2008; Kok and Brockett, 2010).

(1) a lot of $NN_1 \leftrightarrow$ plenty of NN_1
(2) NNP_1 area $\leftrightarrow$ NNP_1 region
(3) NNP_1 wrote $NNP_2 \leftrightarrow NNP_2$ was written by NNP_1

In the basic form of their method, called Model 1, Zhao et al. (2009b) use a log-linear ranker to assign scores to candidate English paraphrase pairs $\langle e_1, e_2 \rangle$; the ranker uses the alignment probabilities $P(c|e_1)$ and $P(e_2|c)$ as features, along with features that assess the quality of the corresponding alignments. In an extension of their method, Model 2, Zhao et al. consider two English phrases e_1 and e_2 as paraphrases, if they are often aligned to two Chinese phrases c_1 and c_2, which are themselves paraphrases according to Model 1 (with English used as the pivot language). Again, a log-linear ranker assigns a score to each $\langle e_1, e_2 \rangle$ pair, now with $P(c_1|e_1)$, $P(c_2|c_1)$, and $P(e_2|c_1)$ as features, along with similar features for alignment quality. In a further extension, Model 3, all the candidate phrase pairs $\langle e_1, e_2 \rangle$ are collectively treated as a monolingual parallel corpus. The phrases of the corpus are aligned, as when aligning a bilingual parallel corpus, and additional features, based on the alignment, are added to the log-linear ranker, which again assigns a score to each $\langle e_1, e_2 \rangle$.

The resulting paraphrasing rules $e_1 \leftrightarrow e_2$ typically contain short phrases (up to four or five words excluding slots) on each side; hence, they can be used to rewrite only parts of longer sentences. Given an input (source) sentence S, we generate candidate paraphrases by applying rules whose left or right hand side matches any part of S. For example, rule (1) matches the source sentence (4); hence, (4) can be rewritten as the candidate paraphrase (5).[3]

(4) S: He had $\underline{\text{a lot of } [_{NN_1}\text{admiration}]}$ for his job.
(5) C: He had $\underline{\text{plenty of } [_{NN_1}\text{admiration}]}$ for his job.

Several rules may apply to S; for example, they may rewrite different parts of S, or they may replace the same parts of S by different phrases. We allow all possible combinations of applicable rules to apply to S, excluding combinations that include rules rewriting overlapping parts of S.[4] To avoid generating too many candidates (C), we use only the 20 rules (that apply to S) with the highest scores. Zhao et al. actually associate each rule with three scores. The first one, hereafter called r_1, is the Model 1 score, and the other two, r_2 and r_3, are the forward and backward alignment probabilities of Model 3; see Zhao et al. (2009b) for details. We use the average of the three scores, hereafter r_4, when generating candidates.

Unfortunately, Zhao et al.'s scores reflect the overall quality of each rule, without considering the context of the particular S where the rule is applied. Szpektor et al. (2008) point out that, for example, a rule like "X acquire Y" $\leftrightarrow$ "X buy Y" may work well in many contexts, but not in "Children acquire language quickly". Similarly, "X charged Y with" $\leftrightarrow$ "X accused Y of" should not be applied to sentences about charging batteries. Szpektor et al. propose, roughly speaking, to associate each rule with a model of the contexts where the rule is applicable, as well as models of the expressions that typically fill its slots, in order to be able to assess the applicability of each rule in specific contexts. The rules that we use do not have associated models of this kind, but we follow Szpektor et al.'s idea of assessing the applicability of each rule in each particular context, when ranking candidates, as discussed below.

3 A dataset of candidate paraphrases

Our generate and rank method relies on existing large collections of paraphrasing rules to generate candidate paraphrases. Our main contribution is in the ranking of the candidates. To be able to evaluate the performance of different rankers in the task we are concerned with, we first constructed an evaluation dataset that contains pairs $\langle S, C \rangle$ of source (input) sentences and candidate paraphrases, and we asked human judges to assess the degree to which the C of each pair was a good paraphrase of S.

We selected randomly 75 source (S) sentences from the AQUAINT corpus, such that at least one of the paraphrasing rules applied to each S.[5] For each S, we generated candidate Cs using Zhao et al.'s rules, as discussed in Section 2. This led to 1,935 $\langle S, C \rangle$ pairs, approx. 26 pairs for each S. The pairs were given to 13 judges other than the authors.[6] Each judge evaluated approx. 148 (different) $\langle S, C \rangle$

[3] We use Stanford's POS tagger, MaxEnt classifier, and dependency parser; see http://nlp.stanford.edu/.

[4] A possible extension, which we have not explored, would be to recursively apply the same process to the resulting Cs.

[5] The corpus is available from the LDC (LDC2002T31).

[6] The judges were fluent, but not native English speakers.

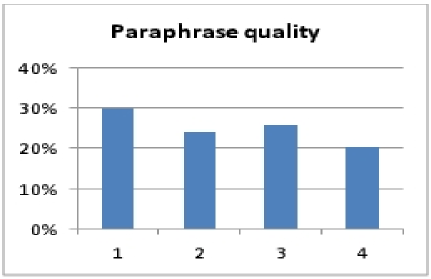

Figure 1: Distribution of overall quality scores in the evaluation dataset (1 = totally unacceptable, 4 = perfect).

pairs; each of the 1,935 pairs was evaluated by one judge. The judges were asked to provide grammaticality, meaning preservation, and overall paraphrase quality scores for each $\langle S, C \rangle$ pair, each score on a 1–4 scale (1 for totally unacceptable, 4 for perfect); guidelines and examples were also provided.

Figure 1 shows the distribution of the overall quality scores in the 1,935 $\langle S, C \rangle$ pairs of the evaluation dataset; the distributions of the grammaticality and meaning preservation scores are similar. Notice that although we used only the 20 applicable paraphrasing rules with the highest scores to generate the $\langle S, C \rangle$ pairs, less than half of the candidate paraphrases (C) were considered good, and approximately only 20% perfect. In other words, applying paraphrasing rules (even only those with the 20 best scores) to each input sentence S and randomly picking one of the resulting candidate paraphrases C, without any further filtering (or ranking) of the candidates, would on average produce unacceptable paraphrases more frequently than acceptable ones. Hence, the role of the ranking component is crucial.

We also measured inter-annotator agreement by constructing, in the same way, 100 additional $\langle S, C \rangle$ pairs (other than the 1,935) and asking 3 of the 13 judges to evaluate all of them. We measured the mean absolute error, i.e., the mean absolute difference in the judges' scores (averaged over all pairs of judges) and the mean (over all pairs of judges) K statistic (Carletta, 1996). In the overall scores, K was 0.64, which is in the range often taken to indicate substantial agreement (0.61–0.80).[7] Agreement was higher for grammaticality ($K = 0.81$), and lower ($K = 0.59$) for meaning preservation. Table 1 shows that the mean absolute difference in the annotators' scores was $\frac{1}{5}$ to $\frac{1}{4}$ of a point.

	mean abs. diff.	K-statistic
grammaticality	0.20	0.81
meaning preserv.	0.26	0.59
overall quality	0.22	0.64

Table 1: Inter-annotator agreement when manually evaluating candidate paraphrases.

Several judges commented that they had trouble deciding to what extent the overall quality score should reflect grammaticality or meaning preservation. They also wondered if it was fair to consider as perfect candidate paraphrases that differed in only one or two words from the source sentences, i.e., candidates with low diversity. These comments led us to ignore the judges' overall quality scores in some experiments, and to use a weighted average of grammaticality, meaning preservation, and (automatically measured) diversity instead, with different weight combinations corresponding to different application requirements, as discussed further below.

In the same way, 1,500 more $\langle S, C \rangle$ pairs (other than the 1,935 and the 100, not involving previously seen Ss) were constructed, and they were evaluated by the first author. The 1,500 pairs were used as a training dataset in experiments discussed below. Both the 1,500 training and the 1,935 evaluation (test) pairs are publicly available.[8] We occasionally refer to the training and evaluation datasets as a single dataset, but they are clearly separated.

4 Ranking candidate paraphrases

We now discuss the ranking component of our method, which assesses the candidate paraphrases.

4.1 Features of the ranking component

Each $\langle S, C \rangle$ pair is represented as a feature vector. To allow the ranking component to assess the degree to which a candidate C is grammatical, or at least as grammatical as the source S, we include in the feature vectors the language model scores of S, C, and the difference between the two scores. We use a 3-gram language model trained on approximately

[7] It is also close to 0.67, which is sometimes taken to be a cutoff for substantial agreement in computational linguistics.

[8] See the paper's supplementary material.

6.5 million sentences of the AQUAINT corpus.[9] To allow the ranker to consider the (context-insensitive) quality scores of the rules that generated C from S, we also include as features the highest, lowest, and average r_1, r_2, r_3, and r_4 scores (Section 2) of these rules, 12 features in total.

The features discussed so far are similar to those employed by Zhao et al. (2009a) in the only comparable paraphrase generation method we are aware of that uses paraphrasing rules. That method, hereafter called ZHAO-RUL, uses the language model score of C and scores similar to r_1, r_2, r_3 in a log-linear model.[10] The log-linear model of ZHAO-RUL is used by an SMT-like decoder to identify the transformations (applications of rules) that produce the (hopefully) best paraphrase. By contrast, we first generate a large number of candidates using the paraphrasing rules, and we then rank them. Unfortunately, we did not have access to an implementation of ZHAO-RUL to compare against, but below we compare against another paraphraser proposed by Zhao et al. (2010), hereafter called ZHAO-ENG, which uses multiple machine translation engines and pivot languages, instead of paraphrasing rules, and which Zhao et al. found to outperform ZHAO-RUL.

To further help the ranking component assess the degree to which C preserves the meaning of S, we also optionally include in the vectors of the $\langle S, C\rangle$ pairs the features of an existing paraphrase recognizer (Malakasiotis, 2009) that obtained the best published results (Androutsopoulos and Malakasiotis, 2010) on the widely used MSR paraphrasing corpus.[11] Most of the recognizer's features are computed by using nine similarity measures: Levenshtein, Jaro-Winkler, Manhattan, Euclidean, and n-gram ($n = 3$) distance, cosine similarity, Dice, Jaccard, and matching coefficients, all computed on tokens; consult Malakasiotis (2009) for details. For each $\langle S, C\rangle$ pair, the nine similarity measures are applied to ten different forms $\langle s_1, c_1\rangle, \ldots, \langle s_{10}, c_{10}\rangle$ of $\langle S, C\rangle$, described below, leading to 90 features.

$\langle s_1, c_1\rangle$: The original forms of S and C.
$\langle s_2, c_2\rangle$: S and C with tokens replaced by stems.
$\langle s_3, c_3\rangle$: S and C, with tokens replaced by POS tags.
$\langle s_4, c_4\rangle$: S and C, tokens replaced by soundex codes.[12]
$\langle s_5, c_5\rangle$: S and C, but having removed non-nouns.
$\langle s_6, c_6\rangle$: As previously, but nouns replaced by stems.
$\langle s_7, c_7\rangle$: As previously, nouns replaced by soundex.
$\langle s_8, c_8\rangle$: S and C, but having removed non-verbs.
$\langle s_9, c_9\rangle$: As previously, but verbs replaced by stems.
$\langle s_{10}, c_{10}\rangle$: As previously, verbs replaced by soundex.

When constructing all ten forms $\langle s_i, c_i\rangle$ of $\langle S, C\rangle$, synonyms (in any WordNet synset) are treated as identical words. Additional variants of some of the 90 features compare a sliding window of some of the s_i forms to the corresponding c_i forms (or vice versa), adding 40 more features; see Malakasiotis (2009). Two more Boolean features indicate the existence or absence of negation in S or C, respectively; and another feature computes the ratio of the lengths of S and C, measured in tokens. Finally, three additional features compare the dependency trees of S and C:

$$R_S = \frac{|\text{common dependencies of } S, C|}{|\text{dependencies of } S|}$$

$$R_C = \frac{|\text{common dependencies of } S, C|}{|\text{dependencies of } C|}$$

$$F_{\beta=1} = \frac{2 \cdot R_S \cdot R_C}{R_S + R_C}$$

The recognizer's features are 136 in total.[13] Hence, the full feature set of our paraphraser's ranking component comprises 151 features.

[9] We use SRILM; see `http://www-speech.sri.com/`.

[10] Application-specific features are also included, which can be used, for example, to favor paraphrases that are shorter than the input in sentence compression (Knight and Marcu, 2002; Clarke and Lapata, 2008). Similar features could also be added to application-specific versions of our method.

[11] The MSR corpus contains pairs that are paraphrases or not. It is a benchmark for paraphrase recognizers, not generators. It provides only one paraphrase (true or false) of each source, and few of the true paraphrases can be obtained by the rules we use.

[12] The Soundex algorithm maps English words to alphanumeric codes, so that words with the same pronunciations receive the same codes, despite spelling differences; see `http://en.wikipedia.org/wiki/Soundex`.

[13] Malakasiotis (2009) shows that although there is a lot of redundancy in the recognizer's feature set, the full feature set still leads to better paraphrase recognition results, compared to subsets constructed via feature selection with hill-climbing or beam search. The same paper reports that the recognizer performs almost as well without the last three features, which may not be available in languages with no reliable dependency parsers. Notice, also, that the recognizer does not use paraphrasing rules.

4.2 Learning rate with a MaxEnt classifier

To obtain a first indication of whether or not a ranking component equipped with the features discussed above could learn to distinguish good from bad candidate paraphrases, and to investigate if our training dataset is sufficiently large, we initially experimented with a Maximum Entropy classifier (with the 151 features) as the ranking component. This initial version of the ranking component, called ME-REC, was trained on increasingly larger parts of the training dataset of Section 3, and it was always evaluated on the entire test dataset of that section. For simplicity, we used only the judges' overall quality scores in these experiments, and we treated the problem as one of binary classification; overall quality scores of 1 and 2 where conflated to a negative category, and scores of 3 and 4 to a positive category.

Figure 2 plots the error rate of ME-REC, computed both on the test set and the encountered training subset. The error rate on the training instances a learner has encountered is typically lower than the error rate on the test set (unseen instances); hence, the former error rate can be seen as a lower bound of the latter. ME-REC shows signs of having reached its lower bound when the entire training dataset is used, suggesting that the training dataset is sufficiently large. The baseline (BASE) of Figure 2 uses only a threshold on the average r_4 (Section 2) of the rules that turned S into C. If the average r_4 is higher than the threshold, the $\langle S, C \rangle$ pair is classified in the positive class, otherwise in the negative one. The threshold was tuned by experimenting on a separate tuning dataset. Clearly, ME-REC outperforms the baseline, which uses only the average (context-insensitive) scores of the applied paraphrasing rules.

4.3 Experiments with an SVR ranker

As already noted, when our dataset were constructed the judges felt it was not always clear to what extent the overall quality scores should reflect meaning preservation or grammaticality; and they also wondered if the overall quality scores should have also taken into consideration diversity. To address these concerns, in the experiments described in this section (and the remainder of the paper) we ignored the judges' overall scores, and we used a weighted average of the grammaticality, meaning preservation,

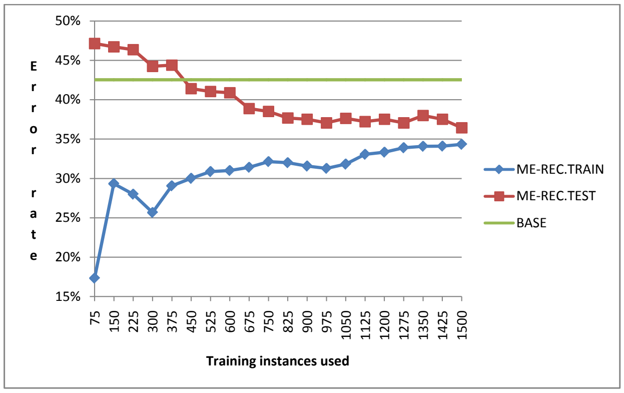

Figure 2: Learning curves of a Maximum Entropy classifier used as the ranking component of our method.

and diversity scores instead; the grammaticality and meaning preservation scores were those provided by the judges, while diversity was automatically computed as the edit distance (Levenshtein, computed on tokens) between S and C. Stated otherwise, the correct score $y(x_i)$ of each training or test instance x_i (i.e., of each feature vector of an $\langle S, C \rangle$ pair) was taken to be a linear combination of the grammaticality score $g(x_i)$, the meaning preservation score $m(x_i)$, and the diversity $d(x_i)$, as in Equation (6), where $\lambda_3 = 1 - \lambda_1 - \lambda_2$.

$$y(x_i) = \lambda_1 \cdot g(x_i) + \lambda_2 \cdot m(x_i) + \lambda_3 \cdot d(x_i) \quad (6)$$

We believe that the λ_i weights should in practice be application-dependent. For example, when paraphrasing user queries to a search engine that turns them into bags of words, diversity and meaning preservation may be more important than grammaticality; by contrast, when paraphrasing the sentences of a generated text to avoid repeating the same expressions, grammaticality is very important. Hence, generic paraphrase generators, like ours, intended to be useful in many different applications, should be evaluated for many different combinations of the λ_i weights. Consequently, in the experiments of this section we trained and evaluated the ranking component of our method (on the training and evaluation part, respectively, of the dataset of Section 3) several times, each time with a different combination of $\lambda_1, \lambda_2, \lambda_3$ values, with the values of each λ_i ranging from 0 to 1 with a step of 0.2.

We employed a Support Vector Regression (SVR) model in the experiments of this section, instead of

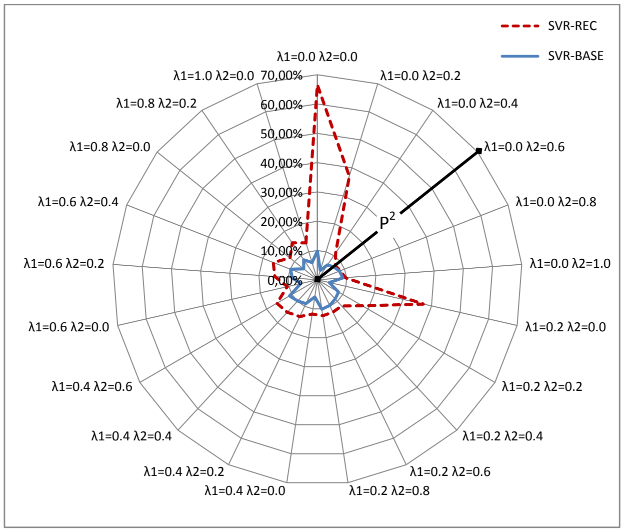

Figure 3: Performance of our method's SVR ranking component with (SVR-REC) and without (SVR-BASE) the additional features of the paraphrase recognizer.

a classifier, given that the $y(x_i)$ scores that we want to predict are real values.[14] An SVR is very similar to a Support Vector Machine (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000; Joachims, 2002), but it is trained on examples of the form $\langle x_i, y(x_i) \rangle$, where $x_i \in \mathbb{R}^n$ and $y(x_i) \in \mathbb{R}$, and it learns a ranking function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ that is intended to return $f(x_i)$ values as close as possible to the correct ones $y(x_i)$, given feature vectors x_i. In our case, the correct $y(x_i)$ values were those of Equation (6). We call SVR-REC the SVR ranker with all the 151 features of Section 4.2, and SVR-BASE the SVR ranker without the 136 features of the paraphrase recognizer.

We used the squared correlation coefficient ρ^2 to evaluate SVR-REC against SVR-BASE.[15] The ρ^2 coefficient shows how well the scores returned by the SVR are correlated with the desired scores $y(x_i)$; the higher the ρ^2 the higher the agreement. Figure 3 shows the experimental results. Each line from the diagram's center represents a different experimental setting, i.e., a different combination of λ_1 and λ_2; recall that $\lambda_3 = 1 - \lambda_1 - \lambda_2$. The distance of a method's curve from the center is the method's ρ^2 for that setting. The farther a point is from the center the higher ρ^2 is; hence, methods whose curves are closer to the diagram's outmost perimeter are better. Clearly, SVR-REC (which includes the recognizer's features) outperforms SVR-BASE (which relies only on the language model and the scores of the rules).

The two peaks of SVR-REC's curve are when λ_3 is very high (1 or 0.8), i.e., when $y(x_i)$ is dominated by the diversity score; in these cases, SVR-REC is at a clear advantage, since it includes features for surface string similarity (e.g., Levenshtein distance measured on $\langle s_1, c_1 \rangle$), which in effect measure diversity, unlike SVR-BASE. Even when λ_1 is very high (1 or 0.8), i.e., when all or most of the weight is placed on grammaticality, SVR-REC outperforms SVR-BASE, indicating that the extra features in SVR-REC also contribute towards assessing grammaticality; by contrast SVR-BASE relies exclusively on the language model for grammaticality. Unfortunately, when λ_2 is very high (1 or 0.8), i.e., when all or most of the weight is placed on meaning preservation, there is no or very small difference between SVR-REC and SVR-BASE, suggesting that the extra features of the paraphrase recognizer are not as useful to the SVR, when assessing meaning preservation, as we would have hoped. Nevertheless, SVR-REC is overall better than SVR-BASE.

We believe that the dataset of Section 3 and the evaluation methodology summarized by Figure 3 will prove useful to other researchers, who may wish to evaluate other ranking components of generate-and-rank paraphrasing methods against ours, for example with different ranking algorithms or features. Similar datasets of candidate paraphrases can also be created using different collections of paraphrasing rules.[16] The same methodology can then be used to evaluate ranking components on those datasets.

5 Comparison to the state of the art

Having established that SVR-REC is a better configuration of our method's ranker than SVR-BASE, we

[14] Additional experiments confirmed that the SVR performs better than ME-REC as the ranking component. We use the SVR implementation of LIBSVM, available from http://www.csie.ntu.edu.tw/∼cjlin/libsvm/, with an RBF kernel and default settings. All the features are normalized in $[-1, 1]$, when using SVR or ME-REC.

[15] If n is the number of test pairs, $f(x_i)$ the score returned by the SVR for the i-th pair, and $y(x_i)$ the correct score, then ρ^2 is:

$$\frac{(n \sum_{i=1}^{n} f(x_i) y_i - \sum_{i=1}^{n} f(x_i) \sum_{i=1}^{n} y(x_i))^2}{(n \sum_{i=1}^{n} f(x_i)^2 - (\sum_{i=1}^{n} f(x_i))^2)(n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y(x_i))^2)}$$

[16] See Androutsopoulos and Malakasiotis (2010) for pointers.

proceed to investigate how well our overall generate-and-rank method (with SVR-REC) compares against a state of the art paraphrase generator.

As already mentioned, Zhao et al. (2010) recently presented a method (we call it ZHAO-ENG) that outperforms their previous method (Zhao et al., 2009a), which used paraphrasing rules and an SMT-like decoder (we call that previous method ZHAO-RUL). Given an input sentence S, ZHAO-ENG produces candidate paraphrases by translating S to 6 pivot languages via 3 different commercial machine translation engines (treated as black boxes) and then back to the original language, again via 3 machine translation engines (54 combinations). Roughly speaking, ZHAO-ENG then ranks the candidate paraphrases by their average distance from all the other candidates, selecting the candidate(s) with the smallest distance; distance is measured as BLEU score (Papineni et al., 2002).[17] Hence, ZHAO-ENG is also, in effect, a generate-and-rank paraphraser, but the candidates are generated by invoking multiple machine translation engines instead of applying paraphrasing rules, and they are ranked by the average distance measure rather than using an SVR.

An obvious practical advantage of ZHAO-ENG is that it exploits the vast resources of existing commercial machine translation engines when generating candidate paraphrases, which allows it to always obtain large numbers of candidate paraphrases. By contrast, the collection of paraphrasing rules that we currently use does not manage to produce any candidate paraphrases in 40% of the sentences of the New York Times part of AQUAINT, because no rule applies. Hence, in terms of ability to always paraphrase the input, ZHAO-ENG is clearly better, though it should be possible to improve our methods's performance in that respect by using larger collections of paraphrasing rules.[18] A further interesting question, however, is how good the paraphrases of the two methods are, when both methods manage to paraphrase the input, i.e., when at least one paraphrasing rule applies to S. This scenario can be seen as an emulation of the case where the collection of paraphrasing rules is sufficiently large to guarantee that at least one rule applies to any source sentence.

To answer the latter question, we re-implemented ZHAO-ENG, with the same machine translation engines and languages used by Zhao et al. (2010). We also trained our paraphraser (with SVR-REC) on the training part of the dataset of Section 3. We then selected 300 random source sentences S from AQUAINT that matched at least one of the paraphrasing rules, excluding sentences that had been used before. Then, for each one of the 300 S sentences, we kept the single best candidate paraphrase C_1 and C_2, respectively, returned by our paraphraser and ZHAO-ENG. The resulting $\langle S, C_1\rangle$ and $\langle S, C_2\rangle$ pairs were given to 10 human judges. This time the judges assigned only grammaticality and meaning preservation scores (on a 1–4 scale); diversity was again computed as edit distance. Each pair was evaluated by one judge, who was given an equal number of pairs from the two methods, without knowing which method each pair came from. The same judge never rated two pairs with the same S. Since we had no way to make ZHAO-ENG sensitive to $\lambda_1, \lambda_2, \lambda_3$, we trained SVR-REC with $\lambda_1 = \lambda_2 = 1/3$, as the most neutral combination of weights.

Table 2 lists the average grammaticality, meaning preservation, and diversity scores of the two methods. All scores were normalized in $[0, 1]$, but the reader should keep in mind that diversity was computed as edit distance, whereas the other two scores were provided by human judges on a 1–4 scale. The grammaticality score of our method was better than ZHAO-ENG's, and the difference was statistically significant.[19] In meaning preservation, ZHAO-ENG was slightly better, but the difference was not statistically significant. The difference in diversity was larger and statistically significant, with the diversity scores indicating that it takes approximately twice as many edit operations (insert, delete, replace) to turn each source sentence to ZHAO-ENG's paraphrase, compared to the paraphrase of our method.

We note that our method can be tuned, by adjusting the λ_i weights, to produce paraphrases with

[17] We use the version of ZHAO-ENG that Zhao et al. (2010) call "selection-based", since they reported it performs overall better than an alternative decoding-based version.

[18] Recall that the paraphrasing rules we use were extracted from an English-Chinese parallel corpus. Additional rules could be extracted from other parallel corpora, like Europarl (http://www.statmt.org/europarl/).

[19] We used Analysis of Variance (ANOVA) (Fisher, 1925), followed by post-hoc Tukey tests to check whether the scores of the two methods differ significantly ($p < 0.05$).

score (%)	our method	ZHAO-ENG
grammaticality	**90.89**	85.33
meaning preserv.	76.67	78.56
diversity	6.50	**14.58**

Table 2: Evaluation of our paraphrasing method (with SVR-REC) against ZHAO-ENG, using human judges. Results in bold indicate statistically significant differences.

higher grammaticality, meaning preservation, or diversity scores; for example, we could increase λ_3 and decrease λ_1 to obtain higher diversity at the cost of lower grammaticality in the results of Table 2.[20] It is unclear how ZHAO-ENG could be tuned that way.

Overall, our method seems to perform well against ZHAO-ENG, despite the vastly larger resources of ZHAO-ENG, provided of course that we limit ourselves to source sentences to which paraphrasing rules apply. It would be interesting to investigate in future work if our method's coverage (sentences it can paraphrase) can increase to ZHAO-ENG's level by using larger collections of paraphrasing rules. It would also be interesting to combine the two methods, perhaps by using SVR-REC (without features for the quality scores of the rules) to rank candidate paraphrases generated by ZHAO-ENG.

6 Conclusions and future work

We presented a generate-and-rank method to paraphrase sentences. The method first produces candidate paraphrases by applying existing paraphrasing rules extracted from parallel corpora, and it then ranks (or classifies) the candidates to keep the best ones. The ranking component considers not only the context-insensitive quality scores of the paraphrasing rules that produced each candidate, but also features intended to measure the extent to which the rule applications preserve grammaticality and meaning in the particular context of the input sentence, as well as the degree to which the resulting candidate differs from the input sentence (diversity).

Initial experiments with a Maximum Entropy classifier confirmed that the features we use can help a ranking component select better candidate paraphrases than a baseline ranker that considers only the average context-insensitive quality scores of the applied rules. Further experiments with an SVR ranker indicated that our full feature set, which includes features from an existing paraphrase recognizer, leads to improved performance, compared to a smaller feature set that includes only the context-insensitive scores of the rules and language modeling scores. We also propose a new methodology to evaluate the ranking components of generate-and-rank paraphrase generators, which evaluates them across different combinations of weights for grammaticality, meaning preservation, and diversity. The paper is accompanied by a paraphrasing dataset we constructed for evaluations of this kind.

Finally, we evaluated our overall method against a state of the art sentence paraphraser, which generates candidates by using several commercial machine translation systems and pivot languages. Overall, our method performed well, despite the vast resources of the machine translation systems employed by the system we compared against. Our method performed better in terms of grammaticality, equally well in meaning preservation, and worse in diversity, but it could be tuned to obtain higher diversity at the cost of lower grammaticality, whereas it is unclear how the system we compare against could be tuned this way. On the other hand, an advantage of the paraphraser we compared against is that it always produces paraphrases; by contast, our system does not produce paraphrases when no paraphrasing rule applies to the source sentence. Larger collections of paraphrasing rules would be needed to improve our method in that respect.

Apart from obtaining and experimenting with larger collections of paraphrasing rules, it would be interesting to evaluate our method in vivo, for example by embedding it in question answering systems (to paraphrase the questions), in information extraction systems (to paraphrase extraction templates), or in natural language generators (to paraphrase template-like sentence plans). We also plan to investigate the possibility of embedding our SVR ranker in the sentence paraphraser we compared against, i.e., to rank candidates produced by using several machine translation systems and pivot languages, as in ZHAO-ENG.

[20] Additional application-specific experiments confirm that this tuning is possible (Malakasiotis, 2011).

Acknowledgments

This work was partly carried out during INDIGO, an FP6 IST project funded by the European Union, with additional funding from the Greek General Secretariat of Research and Technology.[21]

References

I. Androutsopoulos and P. Malakasiotis. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187.

C. Bannard and C. Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proc. of the 43rd* ACL, pages 597–604, Ann Arbor, MI.

C. Callison-Burch, P. Koehn, and M. Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proc. of* HLT-NAACL, pages 17–24, New York, NY.

C. Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proc. of* EMNLP, pages 196–205, Honolulu, HI, October.

J. Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22:249–254.

J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 1(31):399–429.

M. Collins and T. Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69.

N. Cristianini and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

I. Dagan, B. Dolan, B. Magnini, and D. Roth. 2009. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Lang. Engineering*, 15(4):i–xvii. Editorial of the special issue on Textual Entailment.

P. A. Duboue and J. Chu-Carroll. 2006. Answering the question you wish they had asked: The impact of paraphrasing for question answering. In *Proc. of* HLT-NAACL, pages 33–36, New York, NY.

Ronald A. Fisher. 1925. *Statistical Methods for Research Workers*. Oliver and Boyd.

T. Joachims. 2002. *Learning to Classify Text Using Support Vector Machines: Methods, Theory, Algorithms*. Kluwer.

D. Kauchak and R. Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proc. of* HLT-NAACL, pages 455–462, New York, NY.

K. Knight and D. Marcu. 2002. Summarization beyond sentence extraction: A probalistic approach to sentence compression. *Artif. Intelligence*, 139(1):91–107.

P. Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

S. Kok and C. Brockett. 2010. Hitting the right paraphrases in good time. In *Proc. of* HLT-NAACL, pages 145–153, Los Angeles, CA.

Y. Lepage and E. Denoual. 2005. Automatic generation of paraphrases to be used as translation references in objective evaluation measures of machine translation. In *Proc. of the 3rd Int. Workshop on Paraphrasing*, pages 57–64, Jesu Island, Korea.

N. Madnani and B.J. Dorr. 2010. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387.

N. Madnani, F. Ayan, P. Resnik, and B. J. Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proc. of 2nd Workshop on Statistical Machine Translation*, pages 120–127, Prague, Czech Republic.

P. Malakasiotis. 2009. Paraphrase recognition using machine learning to combine similarity measures. In *Proc. of the Student Research Workshop of* ACL-AFNLP, Singapore.

P. Malakasiotis. 2011. *Paraphrase and Textual Entailment Recognition and Generation*. Ph.D. thesis, Department of Informatics, Athens University of Economics and Business, Greece.

Y. Marton, C. Callison-Burch, and P. Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proc. of* EMNLP, pages 381–390, Singapore.

S. Mirkin, L. Specia, N. Cancedda, I. Dagan, M. Dymetman, and I. Szpektor. 2009. Source-language entailment modeling for translating unknown terms. In *Proc. of* ACL-AFNLP, pages 791–799, Singapore.

R. Nelken and S. M. Shieber. 2006. Towards robust context-sensitive sentence alignment for monolingual corpora. In *Proc. of the 11th* EACL, pages 161–168, Trento, Italy.

S. Padó, M. Galley, D. Jurafsky, and C. D. Manning. 2009. Robust machine translation evaluation with entailment features. In *Proc. of* ACL-AFNLP, pages 297–305, Singapore.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of the 40th* ACL, pages 311–318, Philadelphia, PA.

C. Quirk, C. Brockett, and W. B. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proc. of the Conf. on* EMNLP, pages 142–149, Barcelona, Spain.

[21] Consult http://www.ics.forth.gr/indigo/.

S. Riezler and Y. Liu. 2010. Query rewriting using monolingual statistical machine translation. *Computational Linguistics*, 36(3):569–582.

S. Riezler, A. Vasserman, I. Tsochantaridis, V. Mittal, and Y. Liu. 2007. Statistical machine translation for query expansion in answer retrieval. In *Proc. of the 45th* ACL, pages 464–471, Prague, Czech Republic.

I. Szpektor, I. Dagan, R. Bar-Haim, and J. Goldberger. 2008. Contextual preferences. In *Proc. of* ACL-HLT, pages 683–691, Columbus, OH.

V. Vapnik. 1998. *Statistical learning theory*. John Wiley.

S. Zhao, H. Wang, T. Liu, and S. Li. 2008. Pivot approach for extracting paraphrase patterns from bilingual corpora. In *Proc. of* ACL-HLT, pages 780–788, Columbus, OH.

S. Zhao, X. Lan, T. Liu, and S. Li. 2009a. Application-driven statistical paraphrase generation. In *Proc. of* ACL-AFNLP, pages 834–842, Singapore.

S. Zhao, H. Wang, T. Liu, and Li. S. 2009b. Extracting paraphrase patterns from bilingual parallel corpora. *Natural Language Engineering*, 15(4):503–526.

S. Zhao, H. Wang, X. Lan, and T. Liu. 2010. Leveraging multiple MT engines for paraphrase generation. In *Proceedings of the 23rd* COLING, pages 1326–1334, Beijing, China.

L. Zhou, C.-Y. Lin, and Eduard Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In *Proc. of the Conf. on* EMNLP, pages 77–84.