From Lexical to Semantic Features in Paraphrase Identification

Pedro Fialho
INESC-ID, Lisboa, Portugal
Universidade de Évora, Portugal
pedro.fialho@l2f.inesc-id.pt

Luísa Coheur
INESC-ID, Lisboa, Portugal
Instituto Superior Técnico, Universidade de Lisboa, Portugal
luisa.coheur@l2f.inesc-id.pt

Paulo Quaresma
INESC-ID, Lisboa, Portugal
Universidade de Évora, Portugal
pq@uevora.pt

Abstract
The task of paraphrase identification has been applied to diverse scenarios in Natural Language Processing, such as Machine Translation, summarization, or plagiarism detection. In this paper we present a comparative study on the performance of lexical, syntactic and semantic features in the task of paraphrase identification in the Microsoft Research Paraphrase Corpus. In our experiments, semantic features do not represent a gain in results, and syntactic features lead to the best results, but only if combined with lexical features.

2012 ACM Subject Classification Computing methodologies → Natural language processing; Theory of computation → Support vector machines; Information systems → Near-duplicate and plagiarism detection

Keywords and phrases paraphrase identification, lexical features, syntactic features, semantic features

Digital Object Identifier 10.4230/OASIcs.SLATE.2019.9

Acknowledgements This work was supported by national funds through Fundação para a Ciência e Tecnologia (FCT) with reference UID/CEC/50021/2019, through the international project RAGE with reference H2020-ICT-2014-1/644187 and by FCT’s INCoDe 2030 initiative, in the scope of the demonstration project AIA, “Apoiamento Inteligente a empreendedores (chatbots)”, which also supports the scholarship of Pedro Fialho.

1 Introduction
The task of paraphrase identification consists in deciding if two sentences have the same meaning. It is a popular task in Natural Language Processing, as it can be used in several scenarios. For instance, it can be used for evaluation purposes in Machine Translation: a translation result can be missing a reference, and, still, be a good translation; thus, we should be able to see if it is a paraphrase of some sentence in the reference [24]. In addition, paraphrase identification can also be used by a chatbot that has in its knowledge base a set of pre-defined question/answer pairs. Here, a question submitted by the user needs to be compared with existing questions. If the user question is a paraphrase of an existing question, the system only needs to return the appropriate answer [19]; other applications in which paraphrase identification can help include summarization [21], or plagiarism detection [18].
In many cases, just by comparing the shared lexical elements of two sentences (seen as bags of words) we are able to identify paraphrases. However, in many other cases we need to move to a semantic level to be able to say that two sentences are equivalent. For instance, *Symptoms of influenza include fever and nasal congestion, and Fever and nasal congestion are symptoms of influenza* can be identified as paraphrases by taking advantage of features at a lexical level (for instance, by counting the number of common words). However, the previous sentences and the sentence *A stuffy nose and elevated temperature are signs you may have the flu.* will only be identified as paraphrases if we have access to some semantic information, for instance, if we know that *fever* is similar or equal to *elevated temperature* and the same between *nasal congestion* and *stuffy nose*. Thus, a system with the goal of identifying paraphrases should be able to reason at a semantic level. Unfortunately, some semantic features, such as explicit meaning representations, only exist for some languages. The same happens with syntactic features, although at a less dramatic scale, as syntactic analyzers exist for many languages.

In this paper we present a comparative study on the performance of lexical, syntactic and semantic features for paraphrase identification. To the best of our knowledge, the whole set of features that we use in this work was never employed altogether for paraphrase identification, particularly the ensemble of structural modelling for syntax and explicit whole sentence meaning representations for semantics. Results show that syntactic features lead to the best results, but only if combined with lexical features; semantic features in comparison with lexical features, bring a small improvement to recall, f-measure and accuracy when applied in addition to the lexical features.

This paper is organized as follows: in Section 2 we present Related Work, in Section 3 we describe the features from the different linguistic levels, and, in Section 4 we present the experimental setup. Finally, in sections 5 and 6 we present the obtained results and main conclusions, respectively; in the latter section we also point to future work.

## 2 Related work

As previously mentioned, this work is focused on paraphrase identification. Two sentences are paraphrases of each other when they express equivalent meanings. The difficulty of detecting if two sentences have equivalent meaning varies with the linguistic mechanisms employed in paraphrasing, since a target sentence may employ various lexical and/or syntactic transformations on its source.

Popular features employed in paraphrase identification were primarily designed for machine translation evaluation, such as BLEU [26]. However, many other features have already been applied to paraphrase identification, and there are even toolkits that allow to extract features from different linguistic levels. For instance, HARRY [28] provides lexical features from string similarity metrics applied to various word granularities, and SEMILAR [29] provides sentence to sentence similarity metrics based on techniques such as BLEU. It also provides word to word similarity metrics based on semantic information, as it employs Wordnet [7] and co-occurrence models such as Latent Semantic Analysis [16]. In this work we will take advantage of both these toolkits (along with INESC-ID@ASSIN [8]).

Still in the semantic features domain, explicit meaning representations of sentences can also be compared for paraphrase identification purposes. For instance, in [34] features based on the overlap among semantic representations are used. Examples of meaning representations

----

1 https://examples.yourdictionary.com/examples-of-paraphrasing.html
are Abstract Meaning Representation (from now on AMR) [1] and Discourse Representation Structures [14]. In this work we will use AMR representations of sentences to calculate semantic features, as suggested in [12].

Considering syntactic features, some works (e.g., [33, 23]) take advantage of these structures on paraphrase identification. In these scenarios, the features extracted from structural comparison of parse trees, from constituent or dependency analysis, identify which sub trees are the same (structure wise), and may employ lexical semantics on leaf nodes (which carry the words of the sentence) to weight the importance of a common sub tree.

Typically, approaches for paraphrase identification employ a supervised learning setting, where a model is derived from a training corpus, composed by pairs of sentences labeled with 1 or 0 (for instance) considering that they are or they are not paraphrases, respectively. The Microsoft Research Paraphrase Corpus [6], from now on the MSRP corpus, is a popular choice to train and benchmark such models, since there is a constantly updated ranking of the various systems using it\(^2\). Features from machine translation evaluation achieve competitive results in MSRP, as shown in [18]. Although other publicly available corpora exist, as the paraphrases from Twitter messages [15], or, more recently, the open domain questions from Quora\(^3\), in this paper we will target the MSRP corpus.

### 3 Features from different linguistic levels

We gathered features at the different linguistic levels. In the following we describe these sets.

#### 3.1 Lexical Features

We call lexical features to the ones based on different distance metrics calculated between the lexical elements of a sentence, and assuming that these distances can be computed both at the character or word level. We also assume that words can be transformed in their lexical variants, by applying, for instance, stemming or encoding text into the way it sounds. An example of a lexical feature is the longest common subsequence metric applied to lowercased versions of the sentences in analysis.

Table 1 illustrates some of the lexical features used in this work, where each feature corresponds to the application of the metric on the leftmost column to two sequences, built according to the lexical variants identified in the remaining columns (a detailed explanation of each metric can be found in [8]). Such variants comprise lowercased (L) and stemmed (S) versions of the original (O) text. The cluster (C) and Double Metaphone (DM) variants produce a sequence composed by non verbal codes, which:

- for cluster are binary strings that identify the cluster of each word, according to the Brown clustering algorithm [3] on the Yelp dataset of online reviews\(^4\),
- for DM are the codes of the Double Metaphone algorithm for each word.

The trigrams (T) variant produces a sequence with a different length from the number of words in the original sentence, since it is composed by strings of 3 characters, one for each character in the original text.

---

\(^2\) [https://aclweb.org/aclwiki/Paraphrase_Identification_(State_of_the_art)]

\(^3\) [https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs]

\(^4\) [https://www.yelp.com/dataset/]
Table 1 Combination of features with representations, where O, L, S, C, DM and T correspond to Original, Lowercased, Stemmed, Cluster, Double Metaphone and Trigrams, respectively.

<table>
<thead>
<tr>
<th>Feature</th>
<th>O</th>
<th>L</th>
<th>S</th>
<th>C</th>
<th>DM</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCS</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Edit Distance</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cosine Similarity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Abs Length</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Max Length</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Min Length</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Jaccard</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Soft TF-IDF</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NE Overlap</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>NEG Overlap</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Modal Overlap</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>METEOR</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUGE N</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUGE L</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ROUGE S</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>TER</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NCD</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Numeric</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2 Syntactic Features

In what concerns syntactic features we consider that these features are also based in distances, but between syntactic constituents of the sentence. Thus, similarity scores are computed for pairs of trees, based on the number of common substructures [22]. Here, a tree kernel is applied to a pair of parse trees, to automatically produce the similarity scores. For instance, an adjective attached to a noun corresponds to a sub-tree in the full tree of constituents for a source sentence, and if the tree of the target sentence contains a sub-tree with exactly the same leafs (adjective and noun) and root (the syntactic relation), then a tree kernel would consider 3 fragments in common, meaning that both sentences apply the same adjective to the same noun. Further details on such calculation are found in [22].

3.3 Semantic Features

We follow a broad definition of semantic features as all the features that take advantage of some sort of semantic information, either at the lexical level (for instance, by comparing synonyms of two words) or at the sentence level (for instance, by taking advantage of semantic spaces or explicit meaning representations). Considering the latter, we draw on the previously mentioned AMR [1]. An example AMR for the sentence My drawing was not a picture of a hat., from the AMR corpus for the novel “The Little Prince”, can be seen in Figure 1, as produced by trained annotators [1].

In Figure 1 is shown an AMR rooted at concept picture-01, with 01 indicating an entry in OntoNotes [11] where this concept is defined as the act of displaying something in a picture, such that its ARG1 represents what is displayed, as detailed in the corresponding
Figure 1 AMR example.

PropBank [25] frame 5 in which OntoNotes is based (since the latter is not available for free). Hence, this AMR includes expression *a pictured hat*, negated by setting attribute *polarity* of the root concept to a minus sign.

4 Experimental setup

In the following we present the resources involved in our experiments, and the method for their preparation and usage.

4.1 Corpora

As previously mentioned, we will use the Microsoft Research Paraphrase Corpus [6]. Each example in MSRP is composed of 2 sentences and a positive or negative value (0 or 1) representing whether the sentences are a paraphrase or not. We take as train/test set the usual suggested partitions.

4.2 Gathering Lexical Features

Considering the lexical features, we collect them from the two aforementioned toolkits: INESC-ID@ASSIN, a framework used in the ASSIN competition, and HARRY, a toolkit providing string similarity metrics.

In the INESC-ID@ASSIN framework, language independent metrics are applied to different representations of the original text, such as Double Metaphone codes or character trigrams. The 91 features identified in Table 1 were gathered from the INESC-ID@ASSIN framework.

We also use lexical features extracted from HARRY, which also provides a way of extracting lexical features based on 3 different representations of a text: bytes, bits or words. It contributes with 21 different metrics to apply to each representation, although not all metrics are compatible with all representations. For instance, the Normalized compression distance is only applicable to bits. From HARRY, we obtain 62 features, which include string distances such as the Hamming distance and similarity coefficients such as Jaccard. The complete set of features is described in [28].

4.3 Gathering Syntactic Features

Regarding syntactic features, constituency parse trees are obtained with the Shift-Reduce version of the Stanford parser6. Then, tree kernels are applied to such trees. An efficient approach for structural kernels, and particularly tree kernels, was proposed by [30] in uSVM-TK, an SVM modelling platform based on the SVM-LIGHT engine [13]. This is the

5 http://verbs.colorado.edu/propbank/framesets-english-aliases/picture.html
6 http://nlp.stanford.edu/software/srparser.shtml
chosen learning platform for all our experiments (using tree kernels or not). All the tree kernels available in uSVM-TK were employed, namely “Subtree”, “Subset tree”, “Subset tree considering leaf labels” and “Partial tree kernel” [22].

4.4 Gathering Semantic Features

Taking into consideration semantic features, we used the ones from the already mentioned SEMILAR. From this framework, we gather 9 different features on lexical semantics, such that most correspond to a score on sentence similarity calculated from word to word similarities based on Wordnet, Latent Semantic Analysis or Latent Dirichlet Allocation [2]. The latter two word similarities are based on models provided with SEMILAR, pre-trained on Wikipedia and the TASA corpus as described in [32].

In what concerns explicit meaning representations, we obtain the AMR for the sentences with the JAMR parser [10]. Then, and in order to extract semantic features for the AMR, we use SMATCH [4], a metric that computes the distance between two AMR, with its default configuration (hill-climbing with smart initialization and 4 random restarts), established as best setting in the original SMATCH research.

4.5 Evaluation Metrics

Performance is measured with Precision, Recall, F-measure and Accuracy, except for the comparison with other systems from previously mentioned MSRP rank, where only F-measure and Accuracy are reported.

4.6 Machine Learning kits

We use both uSVM-TK [30] and LIBSVM [5] (from its scikit-learn [27] interface) in our experiments. The former allow us to test syntactic features in a plug and play way. The latter was used just for sanity checking, considering the non-syntactic features, as it does not allows a “plug and play” evaluation of syntactic features.

5 Experiments and results

5.1 The impact of the different features

The best results of applying our feature sets to MSRP are shown in Table 2. By SEMANTICS we understand a feature set containing the SEMILAR and SMATCH features, as opposed to using only one of these semantic feature sets.

As expected, lexical features achieve the best results when the majority of words are common or very similar. Also, as expected, lexical features are almost useless when a paraphrase has low lexical overlap, such as when most words in a target sentence are synonyms of the words in the source sentence. In fact, some lexical features are 0 for all training examples of MSRP, as identified with the Facets tool7. Listing 1 shows an example corresponding to paraphrases from the MSRP test partition that were only correctly identified using semantic features, due to low lexical overlap.

7 https://pair-code.github.io/facets/
Table 2 Evaluation results on MSRP (best of all configurations attempted).

<table>
<thead>
<tr>
<th>Features</th>
<th>Prec</th>
<th>Rec</th>
<th>F</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>lexical</td>
<td>78.79%</td>
<td>85.18%</td>
<td>81.86%</td>
<td>74.90%</td>
</tr>
<tr>
<td>lexical + SEMILAR</td>
<td>78.22%</td>
<td>86.40%</td>
<td>82.10%</td>
<td>74.96%</td>
</tr>
<tr>
<td>lexical + SMATCH</td>
<td>77.98%</td>
<td>85.53%</td>
<td>81.58%</td>
<td>74.32%</td>
</tr>
<tr>
<td>lexical + SEMANTICS</td>
<td>77.44%</td>
<td>85.88%</td>
<td>81.44%</td>
<td>73.97%</td>
</tr>
<tr>
<td>syntax</td>
<td>69.87%</td>
<td>95.46%</td>
<td>80.69%</td>
<td>69.62%</td>
</tr>
<tr>
<td>lexical + syntax</td>
<td>79.90%</td>
<td>86.66%</td>
<td><strong>83.14%</strong></td>
<td><strong>76.63%</strong></td>
</tr>
<tr>
<td>lexical + syntax + SEMILAR</td>
<td>79.44%</td>
<td>86.57%</td>
<td>82.85%</td>
<td>76.17%</td>
</tr>
<tr>
<td>lexical + syntax + SMATCH</td>
<td>79.31%</td>
<td>86.92%</td>
<td>82.94%</td>
<td>76.23%</td>
</tr>
<tr>
<td>lexical + syntax + SEMANTICS</td>
<td>79.61%</td>
<td>86.83%</td>
<td>83.06%</td>
<td>76.46%</td>
</tr>
</tbody>
</table>

Listing 1 Example that was not successful classified in lexical + syntax, but it was successful classified in lexical + syntax + SEMANTICS.

Consumers would still have to get a descrambling security card from their cable operator to plug into the set.

To watch pay television, consumers would insert into the set a security card provided by their cable service.

When syntax is not involved (the first 4 results in Table 2), semantics do not improve the performance of lexical features isolated. Overall, syntactic features in combination with lexical features lead to the best results.

5.2 How do we compare with other systems

In order to compare our results with state-of-the-art systems, Table 3 shows the performance of other systems on the MSRP corpus.

Of particular interest is the result from system [31], which employs neural networks, and performs similarly to our best ensemble of features. Although no feature engineering is needed, we are able to explain our results.

System [9] is the most similar to ours, in that it also employs lexical, syntactic and semantic features in the uSVM-TK platform. Although with fewer features, it achieves better results, as it involves more experiments, additional kernels and an exhaustive configuration of SVM parameters.

5.3 The influence of the Machine Learning toolkit

Finally, experiments were also performed in LIBSVM [5], which implements the SVM decision process in a different manner from SVM-LIGHT. Using LIBSVM for the lexical + SEMANTICS experiment results in F measure of 82.62% and accuracy of 76%. Hence, results improved (previous results were of 81.44% and 73.97%, respectively), which suggest an influence of the SVM implementation.
Table 3 Other systems employing MSRP on similar feature types.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>F</th>
<th>Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>lexical similarity [20]</td>
<td>81.3%</td>
<td>70.3%</td>
</tr>
<tr>
<td>distributional semantics [17]</td>
<td>82.8%</td>
<td>75.7%</td>
</tr>
<tr>
<td>neural networks [31]</td>
<td>83.6%</td>
<td>76.8%</td>
</tr>
<tr>
<td>MT metrics [18]</td>
<td>84.1%</td>
<td>77.4%</td>
</tr>
<tr>
<td>tree and graph kernels [9]</td>
<td>85.2%</td>
<td>79.1%</td>
</tr>
<tr>
<td>our best: lexical + syntax</td>
<td>83.1%</td>
<td>76.6%</td>
</tr>
</tbody>
</table>

6 Conclusion and Future Work

We have presented a study on the contribution of lexical, syntactic and semantic features in paraphrase identification on the MSRP corpus.

Semantic features contribute to a performance enhancement over lexical features isolated (if Precision is not considered), but slightly decreases performance when combined with lexical and syntactic features, although by less than 1%. Best results were achieved by syntactic features in combination with lexical ones. Future work includes balancing the amount of features in vector sets, further exploration of SVM parameters, enrich the set of semantic features, study the behaviour of these features in other corpora, and apply the same approach to the tasks of Semantic Textual Similarity and Recognizing Textual Entailment.

References


From Lexical to Semantic Features in Paraphrase Identification


