A Survey on Paraphrase Recognition

Simone Magnolini
University of Brescia
Fondazione Bruno Kessler
magnolini@fbk.eu

Abstract. Paraphrase Recognition is a task of growing interest in the natural language process (NLP) research during the last years. This task aims to detect if two sentences have the same meaning. Paraphrase relationship described in this work is not the definition given by the common knowledge, but is more natural language oriented since it is driven by a lot of background human knowledge. This type of relation can be used as a support for many other NLP applications, such as Question Answering, Multi-document Summarization and for Machine Translation too. This paper presents an overview of the different phenomena that lead to paraphrase, of the methods used for Paraphrase Recognition and of different data-sets used for the task evaluation and of the main issue still opened.

1 Introduction

Paraphrase is an important phenomenon that can be used to improve many other NLP task. Possible ways to face paraphrase are recognition, generation or extraction; in this paper we'll focus only on the first one, but it's easy to show that they are strongly connected. For example with a system that can recognize paraphrase is possible to improve the quality of a paraphrase generator, the first system can choose the best paraphrase among the ones proposed by the generator. The same is true if we have to validate a candidate given by a paraphrase extractor; we can assume that every improvement in one of these tasks will effect also the others.

Before starting to describe the task is useful to give some examples of the possible application of paraphrasing. In question answering (QA) paraphrase can be use in both directions, first is possible that the QA system needs a paraphrase of the original question to find the right answer. A further paraphrase cold be also possible to present the answer. In machine translation (MT) paraphrase can be used to improve the quality of a translation, especially to cover missing expressions or words that the system didn't learn during the training phase [10]. Another important task that use intensively paraphrase recognition technique is plagiarism identification [19], in this task it's request to find ideas and sentences that are paraphrase of others (the original ones), an intense use of synonyms and syntactic modifications can generate very challenging data-sets.

Aim of this paper is to give a wide overview of the problems and of the most challenging part of the task, but we avoid to go deep into details or technical aspects. We begin our survey with a definition of paraphrase, trying to stress the critical aspect related to the task. The second part is focused on the different approach used for paraphrase recognition, for every type of technique we present a paper that use this approach; the description of every system is not the goal of this paper. The third part presents some issue of the paraphrase data-sets.

2 Paraphrase

A definition of *paraphrase* of a sentence, according to the common knowledge, is another sentence with the same meaning of using different words. This definition introduces two important aspects: *same meaning* and *different words*. These two concepts are quite intuitive but difficult to formalize. For example, in [2], three different sentences are proposed:

(1) Wonderworks Ldt. constructed the new bridge.

(2) The new bridge was constructed by Wonderworks Ldt.

(3) Wonderworks Ldt. is the constructor of the new bridge.

Only the (1) and the (2) are actually paraphrase, but many people accept (3) as a paraphrase, too. The idea that in (3) the bridge cold not be finished is usually ignored, we usually accept a little decrease (or increase) of information, if not too large. In [7] is introduced the concept of *"quasi-paraphrases"* to better describe the notion in linguistic, and to distinguish it from the logical definition. Another definition of paraphrase is derived from [2], two sentences T_1, T_2 are paraphrase if T_1 entails T_2 and T_2 entails T_1.

2.1 Paraphrasing phenomena classification

Many linguistic phenomena lead to paraphrase, in [7] are listed 25 possible type of *"substitution"* that maintain the meaning of the sentence inside the boundaries of the *quasi-paraphrase*. This analysis has a linguistic point of view and not a computational approach to the task, this grants a wider classification that includes also rare phenomena that don't compare in the most used data-sets, but that are common in the spoken language. According to this analysis is interesting to note that the type of paraphrase are not uniformly used, but that three of them collect more than the 75% of the examples in both the data-sets taken into consideration in [7]. The types (with an example) are:

Synonym substitution: ”He *quickly* leaves the room” ≡ ”He *speedily* leaves the room”

Function word variation: ”This letter is very important *to* your admission” ≡ ”This letter is very important *for* your admission”

External knowledge: ”*Obama* was named the 2009 Nobel Peace Prize laureate” ≡ ”*The President of the United States* was named the 2009 Nobel Peace Prize laureate”

In [21] is proposed a more structured classification that takes into consideration not only the types of substitution, but divides the 24 types in 7 sub-classes and in 5 classes. In the paper is pointed the attention on the coexistence of more than one phenomena in a paraphrase.
The presence of one modification is not enough to be sure that two sentences are paraphrase linked, for example:

I like my dog

I like dogs

It’s a general substitution, but the meaning is different.
At the moment the classification of different paraphrase is not used for recognizing tasks, this is due to the leak of annotated data-sets with this kind of information, and to the absence of a unique classification for the paraphrase. Understanding the phenomena that origin the paraphrase is important for two main goals: add features to the systems that can be used and find the boundaries of the task in order to obtain better training set.

3 Different Approaches

The paraphrase recognition is used as a part, sometimes as the main part, of different NLP systems. The semantic text similarity (STS) is one of them, in this task the system has to measure the similarity of meaning of two sentences in a range from 0 (nothing in common) to 5 (perfect paraphrase). Systems that take part at this challenge use, sometimes, as training and test sets paraphrase data-sets, also the STS data-set is sometimes considered a paraphrase data-set. An overview of the different approach to the task will include also systems designed for the STS.
Others common approaches to this task are textual entailment systems, since paraphrase can be seen as a double entailment.

Table 1. State of the art for paraphrase as show on *http://aclweb.org/aclwiki/index.php*. F-score is the harmonic mean of precision and recall

Algorithm	Description	Supervision	Accuracy	F
Vector Based Similarity (Baseline)	cosine similarity with tf-idf weighting	unsupervised	65.4%	75.3%
ESA	explicit semantic space	unsupervised	67.0%	79.3%
KM	combination of lexical and semantic features	supervised	76.6%	79.6%
LSA	latent semantic space	unsupervised	68.8%	79.9%
RMLMG	graph subsumption	unsupervised	70.6%	80.5%
MCS	combination of several word similarity measures	unsupervised	70.3%	81.3%
STS	combination of semantic and string similarity	unsupervised	72.6%	81.3%
SSA	salient semantic space	unsupervised	72.5%	81.4%
QKC	sentence dissimilarity classification	supervised	72.0%	81.6%
ParaDetect	PI using semantic heuristic features	supervised	74.7%	81.8%
SDS	simple distributional semantic space	supervised	73.0%	82.3%
matrixJcn	JCN WordNet similarity with matrix	unsupervised	74.1%	82.4%
FHS	combination of MT evaluation measures as features	supervised	75.0%	82.7%
PE	product of experts	supervised	76.1%	82.7%
WDDP	dependency-based features	supervised	75.6%	83.0%
SHPNM	recursive autoencoder with dynamic pooling	supervised	76.8%	83.6%
MTMETRICS	combination of eight machine translation metrics	supervised	**77.4%**	**84.1%**

3.1 Logic-based approach

In this kind of approach, like in [8], the pair of sentences are mapped into a logic form and then a prover extracts a similarity score based on the operations needed to satisfy both the sentences. In this system background-knowledge doesn't affect the similarity score given by the logic prover, but is used combined with this score for a further elaboration of the pair of sentences. This approach is generally worse that others, but doesn't need knowledge from large corpora.
Another approach is to use background knowledge inside the prover and durin training generate a threshold for the proof found. An example of this approach can be found in [18] for the workshop [1]. This kind of approach is designed for textual entailment, but, as explained, can be used also for paraphrase. Axioms are introduced into the system by a source called eXtended WordNet Knowledge Base (XWN-KB) that grants good results; this shows that some algorithms need also good knowledge base to obtain notable result.

3.2 Vector Space Models approach

In this approach every word is mapped in a vector that contain other words with a score that represents the connection between that word and the others. The quality of the vector depends on the corpora used to generate the model and on the length of the vector [15].
An example of this kind of mapping is Word2Vec [15], this tool is used in [6] for detecting paraphrase candidates inside a parser. In this paraphrase system the similarity score is calculated with the product of a combination of the components of the vector representations of the sentences. This is not the only possible way to use this model, as it is also possible to sum or subtract the vector to obtain different similarity scores.

3.3 String Similarity approaches

To overcome the problem of transformation from natural language into logic form or into other models a variety of new approaches was developed. In this kind of paraphrase systems the decision is taken just on the analysis of the two texts with a simpler pre-processing. Part-of-speech (POS) tagging and lemmatisation are examples of elaborations used in this kind of approach. The main hypothesis of this kind of approach is that even if in a paraphrase different words are used, a lot of them remain the same (we'll take more into consideration this point during the data-sets analysis). A system may for example count only the lexical overlap, or the edit distance to score the paraphrase distance of two sentence.
The string similarity is used also as a base-line for other systems, like in [20]; a system can take into consideration not only the single token in the two sentences but also the common n-grams [16].

3.4 Syntactic Similarity Approach

It's possible to detect paraphrases not only at semantic level, with a structure similar to bag of words, but analyzing also the syntactic level. An example of this kind of approach may be found in [9] or in [4] (a textual entailment system). In these systems the paraphrase is decided on the number of syntactic rules used to transform a sentence into the other; the rules can be derived from grammatical analysis or be based on a statistical approach.
The number of possible rules is usually quite big, and to detect a non paraphrase case it's needed to try all of them, so some systems can decide to reduce the set of possible transformations to the most probable ones.

Table 2. Results (taken from [16]) for a simple lexical overlap method with various combinations of pre-processing steps for the MSRP test (top four rows) and ULPC (User Language Paraphrase Corpus [14]) test (bottom two rows). The meaning of the abbreviation is explained in section 4

Method	Accuracy	Precision	Recall
Open.Average.P.B.C.U.N.N.	0.7258	0.7705	0.837
Open.Average.P.B.I.U.N.N.	0.7403	0.7538	0.905
Stanford.Maximum.W.B.C.U.N.F.	0.7432	0.76	0.8971
Open.Average.P.B.I.B.I.N.	0.6783	0.6947	0.9207
Stanford.Average.S.B.I.U.N.N.	0.6433	0.6156	0.9267
Open.Average.P.B.I.B.N.F.	0.6072	0.6066	0.8022

3.5 Machine Learning approach

This approach covers a large variety of systems that use different measures listed above as features for machine learning algorithms. For example in [9] not only syntax is taken into consideration but also semantic, named entity recognition, overlap and other features using the ν-Support Vector Regression model (ν-SVR) [17].

This approach is focused not only on the type of similarity relation but also on the impact that every feature may have on the paraphrase relationship. Select the right features is an hard task and can be managed also using heuristic taken from operative research like in [11] where a genetic algorithm is used to select features for a Support Vector Machine (SVM).

3.6 Machine Translation approach

This approach uses large bilingual corpora to detect word or n-grams that are translated in the same way. This idea is also the main assumption used to create the paraphrase database (PPDB) [13]. In these systems are applied ideas or measures already described in the previous points, but with different training set: bilingual corpora. The main advantage is that these kinds of corpora are bigger and easier to obtain that the paraphrase ones.

The system described in [3] shows that this approach is designed for paraphrase extraction. We decide to mention it because with this is possible to create resources used as a feature for recognizing algorithms.

4 Data-sets

In the previous sections of this paper we have stressed some points that still have a vague answer that are: what is the meaning of a sentence and how many words have to be different to define a pair of sentence as a paraphrase? The answers to these questions are, also, in the data-sets; we can notice that many algorithm to paraphrase recognition are supervised, so we can assume that from a computational point of view two sentences have the same meaning when they're tagged like this in a training set. This is quite common for NLP tasks, but is quite interesting to notice that, as discussed in [16], the data-sets for paraphrasing are quite heterogeneous; different annotator, different score. The main corpus for this task is the Microsoft Research Paraphrase corpus (MSRP) [12]: a deep analysis of this corpus and of the others usually used to train and evaluate the systems can be found in [16], we take into consideration only the result on Table 2.
The interesting property of the data-sets is the distribution of the lexical overlap. It's easy to notice that simple overlap systems can obtain good results with easy elaboration. The first letter in a methods name indicates OpenNLP package (O) or Stanford NLP package (S). The second letter indicates the type of normalization for the lexical overlap: average length (A) versus maximum length (M). The remaining indicate: (1) tokens used (P means we compared all tokens, including punctuation; W means we excluded punctuation; C means content words only; S means all words, excluding the stop words), (2) form of the tokens used (W original raw form, B base form, P only words with the same POS and same base form), (3) case sensitivity (S) or insensitivity (I), (4) unigrams (U) or bigrams (B), (5) type of global weight used for each token (I means IDF, E means entropy-based, or N means weight of 1), and (6) type of local weight used (F means word type frequency, N means local weighting of 1).
If we compare the results described in Table 2 with the result of state of the art systems described in Table 1 we can notice that the use of more complex systems grant little improvements. This issue is due to structure and to assumption of the corpora used to train and to evaluate systems, and not to the systems themselves.
This problem is typical also for other NLP tasks as described in [5] for recognizing textual entailment (RTE) task, but to obtain more significant (and challenging) data-sets this trait should be reduced or distributed between paraphrase and not paraphrase sentence pairs.

5 Conclusion

Paraphrasing recognition is a challenging and popular research area, that shares many points with other difficult and useful tasks like STS and RTE. In this pa-

per we have presented an overview, not only about approaches, but also about data-sets and possible applications of paraphrasing recognition.
Important characteristics of the task are: the absence, at the moment, of an algorithm or approach that overwhelms the others; data-sets are strongly influenced by lexical overlap; no linguistic classification is used to face the task. We expect to see a lot of effort on this task, because every improvement on this field can extended also to other important (and maybe more practical) NLP systems.

References

1. *Rte '07: Proceedings of the acl-pascal workshop on textual entailment and paraphrasing*, Stroudsburg, PA, USA, Association for Computational Linguistics, 2007.
2. Ion Androutsopoulos and Prodromos Malakasiotis, *A survey of paraphrasing and textual entailment methods*, arXiv preprint arXiv:0912.3747 (2009).
3. Colin Bannard and Chris Callison-Burch, *Paraphrasing with bilingual parallel corpora*, Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, 2005, pp. 597–604.
4. Roy Bar-Haim, Ido Dagan, Iddo Greental, and Eyal Shnarch, *Semantic inference at the lexical-syntactic level*, PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, vol. 22, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 871.
5. Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo, *The fifth PASCAL recognising textual entailment challenge*, Proceedings of the TAC Workshop on Textual Entailment (Gaithersburg, MD), 2009.
6. Jonathan Berant and Percy Liang, *Semantic parsing via paraphrasing*, Proceedings of ACL, 2014.
7. Rahul Bhagat and Eduard Hovy, *What is a paraphrase?*, Computational Linguistics **39** (2013), no. 3, 463–472.
8. Eduardo Blanco and Dan I Moldovan, *A logic prover approach to predicting textual similarity.*, FLAIRS Conference, 2013.
9. Davide Buscaldi, Joseph Le Roux, Jorge J García Flores, Adrian Popescu, et al., *Lipn-core: Semantic text similarity using n-grams, wordnet, syntactic analysis, esa and information retrieval based features*, * SEM 2013 (2013).
10. Chris Callison-Burch, Philipp Koehn, and Miles Osborne, *Improved statistical machine translation using paraphrases*, Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Association for Computational Linguistics, 2006, pp. 17–24.
11. A. CHITRA and ANUPRIYA RAJKUMAR, *Genetic algorithm based feature selection for paraphrase recognition*, International Journal on Artificial Intelligence Tools **22** (2013), no. 02, 1350007.
12. Bill Dolan, Chris Quirk, and Chris Brockett, *Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources*, Proceedings of the 20th international conference on Computational Linguistics, Association for Computational Linguistics, 2004, p. 350.
13. Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch, *Ppdb: The paraphrase database.*, HLT-NAACL, 2013, pp. 758–764.
14. Philip M McCarthy and Danielle S McNamara, *The user-language paraphrase challenge*, Retrieved January **10** (2008), 2008.

15. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781 (2013).
16. Vasile Rus, Rajendra Banjade, and Mihai Lintean, *On paraphrase identification corpora*, Proceeding on the International Conference on Language Resources and Evaluation (LREC 2014), 2014.
17. Bernhard Scholkopf, Peter L Bartlett, Alex J Smola, and Robert Williamson, *Shrinking the tube: a new support vector regression algorithm*, Advances in neural information processing systems (1999), 330–336.
18. Marta Tatu and Dan Moldovan, *Cogex at rte3*, Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Association for Computational Linguistics, 2007, pp. 22–27.
19. Özlem Uzuner, Boris Katz, and Thade Nahnsen, *Using syntactic information to identify plagiarism*, Proceedings of the second workshop on Building Educational Applications Using NLP, Association for Computational Linguistics, 2005, pp. 37–44.
20. V Vaishnavi, M Saritha, and RS Milton, *Paraphrase identification in short texts using grammar patterns*, Recent Trends in Information Technology (ICRTIT), 2013 International Conference on, IEEE, 2013, pp. 472–477.
21. Marta Vila, M Antònia Martí, and Horacio Rodríguez, *Is this a paraphrase? what kind? paraphrase boundaries and typology*, Open Journal of Modern Linguistics **2014** (2014).

This article was processed using the LaTeX macro package with LLNCS style