From lexical to semantic features in paraphrase identification

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

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

Paulo Quaresma
INESC-ID
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 funded by FCT's INCoDe 2030 initiative, in the scope of the demonstration project AIA, "Apoio 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 [25]. 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 [20]; other applications in which paraphrase identification can help include summarization [22], or plagiarism detection [19].

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,

8th Symposium on Languages, Applications and Technologies (SLATE 2019).
Editors: Ricardo Rodrigues, Jan Janousek, Luís Ferreira, Luísa Coheur, Fernando Batista, and Hugo Gonçalo Oliveira; Article No. 9; pp. 9:1–9:11
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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.*[1] 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 [27]. 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 [29] provides lexical features from string similarity metrics applied to various word granularities, and SEMILAR [30] 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 [17]. 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 [35] features based on the overlap among semantic representations are used. Examples of meaning representations are *Abstract Meaning Representation* (from now on AMR) [1] and *Discourse Representation Structures* [15]. In this work we will use AMR representations of sentences to calculate semantic features, as suggested in [13].

---

[1] https://examples.yourdictionary.com/examples-of-paraphrasing.html

Considering syntactic features, some works (e.g., [34, 24]) 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 [19]. Although other publicly available corpora exist, as the paraphrases from Twitter messages [16], 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.

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 [23]. Here, a tree kernel is

[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/

Feature	O	L	S	C	DM	T
LCS	X	X	X	X	X	
Edit Distance	X	X	X	X	X	
Cosine Similarity	X	X	X	X	X	X
Abs Length	X	X	X	X	X	
Max Length	X	X	X	X	X	
Min Length	X	X	X	X	X	
Jaccard	X	X	X	X	X	X
Soft TF-IDF	X	X	X			
NE Overlap	X	X	X	X	X	X
NEG Overlap	X	X	X	X	X	X
Modal Overlap	X	X	X	X	X	X
METEOR	X	X	X	X	X	
ROUGE N	X	X	X	X	X	
ROUGE L	X	X	X	X	X	
ROUGE S	X	X	X	X	X	
TER	X	X	X	X	X	
NCD	X	X	X	X	X	
Numeric	X	X	X			

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.

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 [23].

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].

```
(p / picture-01 :polarity -
      :ARG0 (p2 / picture
            :ARG1-of (d / draw-01
                  :ARG0 (i / i)))
      :ARG1 (h / hat))
```

Figure 1 AMR example

In Figure 1 is shown an AMR rooted at concept *picture-01*, with *01* indicating an entry in

OntoNotes [12] 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 PropBank [26] 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 [29].

4.3 Gathering Syntactic Features

Regarding syntactic features, constituency parse trees are obtained with the Shift-Reduce version of the Stanford parser[6]. Then, tree kernels are applied to such trees. An efficient approach for structural kernels, and particularly tree kernels, was proposed by [31] in *uSVM-TK*, an SVM modelling platform based on the SVM-LIGHT engine [14]. This is the 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" [23].

[5] `http://verbs.colorado.edu/propbank/framesets-english-aliases/picture.html`
[6] `http://nlp.stanford.edu/software/srparser.shtml`

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 [33].

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* [31] and LIBSVM [5] (from its *scikit-learn* [28] 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 tool[7]. Figure 2 shows an example corresponding to paraphrases from the MSRP test partition that were only correctly identified using semantic features, due to low lexical overlap.

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.

[7] `https://pair-code.github.io/facets/`

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

Table 2 Evaluation results on MSRP (best of all configurations attempted).

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 .

Figure 2 Example that was not successful classified in *lexical + syntax*, but it was successful classified in *lexical + syntax + SEMANTICS*

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 [32], 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.

	F	Acc
lexical similarity [21]	81.3%	70.3%
distributional semantics [18]	82.8%	75.7%
neural networks [32]	83.6%	76.8%
MT metrics [19]	84.1%	77.4%
tree and graph kernels [9]	85.2%	79.1%
our best: lexical + syntax	83.1%	76.6%

Table 3 Other systems employing MSRP on similar feature types.

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

1 Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL: `http://www.aclweb.org/anthology/W13-2322`.

2 David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. URL: `http://dl.acm.org/citation.cfm?id=944919.944937`.

3 Peter F. Brown, Peter V. deSouza, Robert L. Mercer, T. J. Watson, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 1992. URL: `http://aclweb.org/anthology/J92-4003`.

4 Shu Cai and Kevin Knight. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 748–752. The Association for Computer Linguistics, 2013. URL: `http://aclweb.org/anthology/P/P13/P13-2131.pdf`.

5 Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

6 Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, January 2005. URL: `https://www.microsoft.com/en-us/research/publication/automatically-constructing-a-corpus-of-sentential-paraphrases/`.

7 Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

8 Pedro Fialho, Ricardo Marques, Bruno Martins, Luísa Coheur, and Paulo Quaresma. Introducing inesc-id@assin for measuring semantic similarity and recognizing textual entailment. In *Proceedings of the ASSIN Shared Task at the International Conference on the Computational Processing of Portuguese*, Tomar, Portugal, July 2016.

9 Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. Structural representations for learning relations between pairs of texts. In for Computer Linguistics [11], pages 1003–1013. URL: `http://aclweb.org/anthology/P/P15/P15-1097.pdf`.

10 Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1426–1436. Association for Computational Linguistics, 2014.

11 The Association for Computer Linguistics, editor. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. The Association for Computer Linguistics, 2015. URL: `http://aclweb.org/anthology/P/P15/`.

12 Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA, June 2006. Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/N06-2015`.

13 Fuad Issa, Marco Damonte, Shay B. Cohen, Xiaohui Yan, and Yi Chang. Abstract meaning representation for paraphrase detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 442–452. Association for Computational Linguistics, 2018. URL: `http://aclweb.org/anthology/N18-1041`, `doi:10.18653/v1/N18-1041`.

14 Thorsten Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA, 1999.

15 Hans Kamp and Uwe Reyle. *From Discourse to Logic - Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in linguistics and philosophy*. Springer, 1993. URL: `http://dx.doi.org/10.1007/978-94-017-1616-1`, `doi:10.1007/978-94-017-1616-1`.

16 Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234. Association for Computational Linguistics, 2017. URL: `http://aclweb.org/anthology/D17-1126`.

17 Thomas K. Landauer and Susan T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

18 Mihai C. Lintean and Vasile Rus. Measuring semantic similarity in short texts through greedy pairing and word semantics. In G. Michael Youngblood and Philip M. McCarthy, editors, *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference, Marco Island, Florida. May 23-25, 2012*. AAAI Press, 2012. URL: `http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS12/paper/view/4421`.

19 Nitin Madnani, Joel Tetreault, and Martin Chodorow. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American

Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT '12, pages 182–190, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL: `http://dl.acm.org/citation.cfm?id=2382029.2382055`.

20 Jerome L. McClendon, Naja A. Mack, and Larry F. Hodges. The use of paraphrase identification in the retrieval of appropriate responses for script based conversational agents. In William Eberle and Chutima Boonthum-Denecke, editors, *Proceedings of the Twenty-Seventh International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014, Pensacola Beach, Florida, May 21-23, 2014*. AAAI Press, 2014. URL: `http://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS14/paper/view/7793`.

21 Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press, 2006. URL: `http://dl.acm.org/citation.cfm?id=1597538.1597662`.

22 Amita Misra, Brian Ecker, and Marilyn Walker. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287. Association for Computational Linguistics, 2016. URL: `http://www.aclweb.org/anthology/W16-3636`, `doi:10.18653/v1/W16-3636`.

23 Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, ECML'06, pages 318–329, Berlin, Heidelberg, 2006. Springer-Verlag.

24 Alessandro Moschitti. Making tree kernels practical for natural language learning. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–120, 2006. URL: `http://www.aclweb.org/anthology/E06-1015`.

25 Sebastian Pado, Michel Galley, Dan Jurafsky, and Christopher D. Manning. Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 297–305. Association for Computational Linguistics, 2009. URL: `http://www.aclweb.org/anthology/P09-1034`.

26 Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005. URL: `https://doi.org/10.1162/0891201053630264`, `arXiv:https://doi.org/10.1162/0891201053630264`, `doi:10.1162/0891201053630264`.

27 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. URL: `http://www.aclweb.org/anthology/P02-1040`, `doi:10.3115/1073083.1073135`.

28 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

29 Konrad Rieck and Christian Wressnegger. Harry: A tool for measuring string similarity. *J. Mach. Learn. Res.*, 17(1):258–262, January 2016. URL: `http://dl.acm.org/citation.cfm?id=2946645.2946654`.

30 Vasile Rus, Mihai Lintean, Rajendra Banjade, Nobal Niraula, and Dan Stefanescu. Semilar: The semantic similarity toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 163–168, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL: `http://www.aclweb.org/anthology/P13-4028`.

31 Aliaksei Severyn and Alessandro Moschitti. Large-scale support vector learning with structural kernels. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge*

Discovery in Databases: Part III, ECML PKDD'10, pages 229–244, Berlin, Heidelberg, 2010. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=1888339.1888355.

32 Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y. Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 801–809. Curran Associates, Inc., 2011. URL: http://papers.nips.cc/paper/4204-dynamic-pooling-and-unfolding-recursive-autoencoders-for-paraphrase-detection.pdf.

33 Dan Stefanescu, Rajendra Banjade, and Vasile Rus. Latent semantic analysis models on wikipedia and tasa. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA), 2014. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/403_Paper.pdf.

34 Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In for Computer Linguistics [11], pages 1556–1566. URL: http://aclweb.org/anthology/P/P15/P15-1150.pdf.

35 Rob van der Goot and Gertjan van Noord. ROB: using semantic meaning to recognize paraphrases. In Daniel M. Cer, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 40–44. The Association for Computer Linguistics, 2015. URL: http://aclweb.org/anthology/S/S15/S15-2007.pdf.