

Paraphrase Identification with Lexico-Syntactic Graph Subsumption

Vasile Rus, Philip M. McCarthy, Mihai C. Lintean

Danielle S. McNamara and Arthur C. Graesser

University of Memphis

Departments of Computer Science, Psychology, and English

Institute for Intelligent Systems

Memphis, TN 38152, USA

{vrus}@memphis.edu

Abstract

The paper presents a new approach to the problem of paraphrase identification. The new approach extends a previously proposed method for the task of textual entailment. The relationship between paraphrases and entailment is discussed to theoretically justify the new approach. The proposed approach is useful because it uses relatively few resources compared to similar systems yet it produces results similar or better than other approaches to paraphrase identification. The approach also offers significantly better results than two baselines. We report results on a standard data set as well as on a new, balanced data set.

Introduction

Paraphrase is a text-to-text relation between two non-identical text fragments that express the same meaning in different ways. We focus on sentential paraphrases in this article. As an example of a paraphrase we show the two sentences below [from the Microsoft Research Paraphrase Corpus (Dolan, Quirk, & Brockett 2004)], where Text B is a paraphrase of Text A and vice versa.

Text A: *A strong geomagnetic storm was expected to hit Earth today with the potential to affect electrical grids and satellite communications.*

Text B: *A strong geomagnetic storm is expected to hit Earth sometime today and could knock out electrical grids and satellite communications.*

Paraphrases are important in a number of applications. In Natural Language Generation, paraphrases are a method to increase diversity of generated text (Iordanskaja, Kittredge, & Polgere 1991). Paraphrases are useful in Intelligent Tutoring Systems [ITSs; (Graesser *et al.* 2005; McNamara *et al.* 2007)] with natural language input to assess whether student articulated answers to deep questions, e.g. conceptual physics questions, are similar-to/paraphrases-of ideal answers. In Question Answering, multiple answers that are paraphrases of each other could be considered as evidence for the correctness of the answer (Ibrahim, Katz, & Lin 2003).

The paraphrase relation between two texts is related to the entailment relation. Textual entailment is the task of deciding, given two text fragments, whether the meaning of one text is entailed (can be inferred) from another text (Dagan, Glickman, & Magnini 2005). We say that T - the entailing text, entails H - the entailed hypothesis. For instance, the Text *Seeing the huge market potential, Yahoo bought Overture Services, Inc.* entails the Hypothesis *Yahoo took over Overture*. The task is relevant to a large number of applications, including Machine Translation, Question Answering, and Information Retrieval (Dagan, Glickman, & Magnini 2005). A paraphrase can be looked at as a bidirectional entailment relation between the two text fragments. In other words, text A is a paraphrase of text B if and only if A entails B and B entails A. Having a solution to the unidirectional problem of entailment would allow one to extend it to address the paraphrase identification problem.

The paper describes a fully-implemented software system that embeds a newly proposed approach to paraphrase identification. The paraphrase identification approach is based on a previously proposed approach to entailment that uses minimal knowledge resources, as compared to other approaches to entailment (Dagan, Glickman, & Magnini 2004 2005). The entailment approach (Rus *et al.* 2008) only uses lexical, syntactic, synonymy and antonymy information. The synonymy and antonymy information is extracted from a thesaurus, i.e. WordNet (Miller 1995), an online dictionary. No deeper processing, word knowledge or automated reasoning is used. The entailment approach was successfully tested on data from the task of recognizing textual entailment [RTE; (Dagan, Glickman, & Magnini 2004 2005)]. In our approach to entailment, each T-H pair is first mapped into two graphs, one for H and one for T, with nodes representing main concepts and links indicating syntactic dependencies among concepts as encoded in H and T, respectively. An entailment score, $entscore(T, H)$ (see Equation 1), is then computed quantifying the degree to which the T-graph subsumes the H-graph. The score is so defined to be non-symmetric, i.e. $entscore(T, H) \neq entscore(H, T)$. We show in this paper how to extend this approach to handle paraphrases.

The rest of the paper is organized as follows. The next section, *Related Work*, presents previous research on paraphrase identification and textual entailment that are closely

$$\begin{aligned}
entscore(T, H) = & \left(\alpha \times \frac{\sum_{V_h \in V_H} \max_{V_t \in V_T} match(V_h, V_t)}{|V_H|} + \right. \\
& \left. \beta \times \frac{\sum_{E_h \in E_H} \max_{E_t \in E_T} synt_match(E_h, E_t)}{|E_H|} + \gamma \right) \times \\
& \frac{(1 + (-1)^{\#neg-rel})}{2}
\end{aligned} \tag{1}$$

related to our work. The *Approach* section presents in detail our lexico-syntactic approach to paraphrase identification. Following that, the *Summary of Results* describes our experiments on the standard Microsoft Paraphrase Corpus. The *Discussion* section analyzes the results obtained, while *Future Work* presents our plans for the future. *Conclusions* ends the paper.

Related Work

Our work is related to efforts in the areas of paraphrase identification and textual entailment. We briefly describe next such previous efforts to better position our work.

Paraphrase identification is the task of deciding whether two text fragments, usually sentences, are paraphrases of each other. Our approach should not be confused with paraphrase extraction which is the task of collecting pairs of text fragments that are paraphrases of each other from different sources. Paraphrase extraction research uses parallel translations of same source text from a foreign language (Barzilay & McKeown 2001), named entities anchors in candidate sentences, and sequence alignment algorithms to extract paraphrases. Different translations of the same original sentence guarantees the paraphrase relation among the corresponding sentences in the parallel translations. Paraphrase extraction from the web was also successfully attempted (Dolan, Quirk, & Brockett 2004).

Paraphrase identification has been previously explored, most notably by (Kozareva & Montoyo 2006), (Mihalcea, Corley, & Strapparava 2006), and (Qiu, Kan, & Chua 2006), who all used the same standard data set as we did, i.e. the Microsoft Paraphrase Corpus, to evaluate their methods. A direct comparison with their methods is thus possible. (Kozareva & Montoyo 2006) proposed a machine learning approach based on lexical and semantic information, e.g. a word similarity measure based on WordNet. They model the problem of paraphrasing as a classification task. Their model uses a set of linguistic attributes and three different machine learning algorithms (Support Vector Machines [SVM], k-Nearest Neighbors, and Maximum Entropy) to induce classifiers. The classifiers are built in a supervised manner from labeled training data in the Microsoft Paraphrase Corpus. All the attributes that they defined are bidirectional, i.e. they capture sentence similarity in both directions. Three types of experiments were conducted. In one experiment they simply compared the three types of classifiers, in a second experiment they tried different attribute mixtures from their original set of attributes, while in a third experiment they combined several classifiers. The

SVM classifier outperformed the others in all experiments. (Mihalcea, Corley, & Strapparava 2006) introduced several *simple* approaches that rely on corpus-based and knowledge-based word-to-word similarity measures. The measures rely on path or node distance in WordNet, e.g. (Leacock & Chodorow 1998), and statistical distributions of words in large corpora [Latent Semantic Analysis, (Landauer, Foltz, & Laham 1998)]. Further, they proposed a so-called *combined* approach that computes a simple average among the output of the simple approaches. The last related work that we could directly compare our work to is (Qiu, Kan, & Chua 2006). They present a two-step approach. In the first step they identify similarities between two sentences in a possible paraphrase relation. In a second step, dissimilarities among the sentences are detected using machine learning methods. The core idea of their approach is to use predicate argument tuples that capture both lexical and syntactic dependencies among words to find similarities between sentences. Similar to (Kozareva & Montoyo 2006), (Mihalcea, Corley, & Strapparava 2006) used WordNet for the knowledge-based similarity measures. Our method differs from the methods of (Kozareva & Montoyo 2006) and (Mihalcea, Corley, & Strapparava 2006) in several ways. First, we use syntactic information as one of the components of our approach. Second, our lexical component relies on word overlap enhanced with synonymy relations from WordNet as opposed to word-to-word similarity measures. Lastly, our approach incorporates negation handling based on antonymy relations in WordNet. With respect to the method of (Qiu, Kan, & Chua 2006), our approach differs almost in the same way it differs from (Kozareva & Montoyo 2006) and (Mihalcea, Corley, & Strapparava 2006) except the use of syntactic information. (Qiu, Kan, & Chua 2006) do use syntactic information, however in a different way from the way we use it. (Qiu, Kan, & Chua 2006) use paths in syntactic trees as features in modeling dissimilarities between two sentences. In the *Summary of Results* section, we compare the results reported by (Kozareva & Montoyo 2006), (Mihalcea, Corley, & Strapparava 2006), and (Qiu, Kan, & Chua 2006) with the results obtained with our approach.

The task of textual entailment was treated in the recent past in one form or another by research groups ranging from informational retrieval to language processing. In one of the earliest explicit treatments of entailment (Monz & de Rijke 2001) proposed a weighted bag of words approach to entailment. Recently, (Dagan & Glickman 2004) presented a probabilistic approach to textual entailment based on lexico-syntactic structures. (Pazienza, Pennacchiotti, & Zanzotto

2005) use a syntactic graph distance approach for the task of textual entailment. More recently, (Kouylekov & Magnini 2005) approached the entailment task with a tree edit distance algorithm on dependency trees. One distinct feature of our lexico-syntactic approach is the negation handling. None of the above mentioned approaches addressed negation.

Approach

Our solution for paraphrase identification is an extension of a unidirectional approach to entailment. Our approach to recognizing textual entailment is based on the idea of subsumption. In general, an object X subsumes an object Y if X is more general than or identical to Y, or alternatively we say Y is more specific than X. The same idea applies to more complex objects, such as structures of interrelated objects.

Applied to textual entailment, subsumption translates into the following: hypothesis H is entailed from T if and only if T subsumes H. The solution has two phases: (I) map both T and H into graph structures and (II) perform a subsumption operation between the T-graph and H-graph. The processing flow is shown in Figure 1. The approach takes as input two raw text fragments and returns a decision (TRUE or FALSE) indicating whether T entails H.

Phase I: From Text to Graph Representations. The two text fragments involved in a textual entailment decision are initially mapped onto a graph representation. The graph representation we employ is based on the dependency-graph formalism of Mel’cuk (1998). The mapping process has three stages: preprocessing, dependency graph generation, and final graph generation. In the preprocessing stage, the system separates the punctuation from words (tokenization), maps morphological variations of words to their base or root form (lemmatization), assigns part-of-speech labels to each word (tagging), and assesses the inter-relationship of major phrases within the texts (parsing). Additional preprocessing operations are also performed, such as detecting collocations of common nouns (for more details see Rus et al., in press). For instance, *joint venture* is identified as a collocation and thus a single node in the graph is generated for it instead of two, i.e. one for *joint* and one for *venture*. The second stage (dependency graph generation) is the actual mapping from text to the graph representation. This mapping is based on information from parse trees generated during the parsing process. A parse tree groups words in a sentence into phrases and organizes these phrases into hierarchical tree structures from which we can easily detect syntactic dependencies among concepts. The system uses Charniak’s (2000) parser to obtain parse trees and Magerman’s (1994) head-detection rules to obtain the head of each phrase. A dependency tree is generated by linking the head of each phrase to its modifiers in a systematic mapping process. The parser is also used for part-of-speech tagging, i.e. no separate part-of-speech tagger is used in preprocessing. In the third stage (final graph generation), the dependency tree generated in the second stage is refined. The dependency tree encodes exclusively local dependencies (head-modifiers), as opposed to remote dependencies, such as the remote subject relation between *bombers* and *enter* in *The*

bombers had not managed to enter the embassy compounds. Thus, in this stage, the dependency tree is transformed onto a dependency graph by generating remote dependencies and direct relations between content words. Remote dependencies are computed by a naive-Bayes functional tagger (Rus & Desai 2005). Direct relations are generated using a simple procedure that directly links content words by eliminating certain types of intermediate relations, e.g. *mod*. For an example of a direct relation between content words, consider the sentence *I visited the manager of the company*. For this sentence, the modifier (*mod*) dependency between the noun *manager* and its attached preposition *of* is replaced with a direct relation between the prepositional head *manager* and prepositional object *company*. Once graph representations are obtained, a graph subsumption operation is initialized, as described below.

Phase II: Graph Subsumption. The textual entailment problem is mapped into a specific example of graph isomorphism called subsumption (also known as containment). Isomorphism in graph theory addresses the problem of testing whether two graphs are the same. A graph $G = (V, E)$ consists of a set of nodes or vertices V and a set of edges E . Graphs can be used to model the linguistic information embedded in a sentence: vertices represent concepts (e.g., *bombers*, *joint venture*) and edges represent syntactic relations among concepts (e.g., the edge labeled *subj* connects the verb *managed* to its subject *bombers*). The Text (T) entails the Hypothesis (H) if and only if the hypothesis graph is subsumed (or contained) by the text graph. The subsumption algorithm for textual entailment has three major steps: (1) find an isomorphism between V_H (set of vertices of the Hypothesis graph) and V_T (set of vertices of the Text graph); (2) check whether the labeled edges in H, E_H , have correspondents in E_T ; (3) compute score. Step 1 is more than a simple word-matching method because if a vertex in H does not have a direct correspondent in T, a thesaurus is used to find all possible synonyms for vertices in T. In addition, vertices in H have different priorities, such as the fact that head words are more important than modifiers. Step 2 takes each relation in H and checks its presence in T. The checking is augmented with relation equivalences among linguistic phenomena such as possessives, and linking verbs (e.g. *be*, *have*). For instance, *tall man* would be equivalent to *man is tall*. Lastly, a normalized score for vertices and edge mapping is computed. The score for the entire entailment is the weighted sum of each individual vertex and edge matching score (see Equation 1). The weights for the vertex and edge matching scores are given by the parameters α and β , respectively. γ is the free term which can be used to bias the score towards making more optimistic decisions (positive values for γ) or not (negative values for γ).

Negation. We also handle two broad types of negation: explicit and implicit. Explicit negation is indicated by particles such as: *no*, *not*, *neither ... nor* and their shortened forms *n’t*. Implicit negation is present in text via deeper lexico-semantic relations among different linguistic expressions, the most obvious example is the *antonymy* relation among words which can be retrieved from WordNet (Miller 1995). Negation is regarded as a feature of both Text and

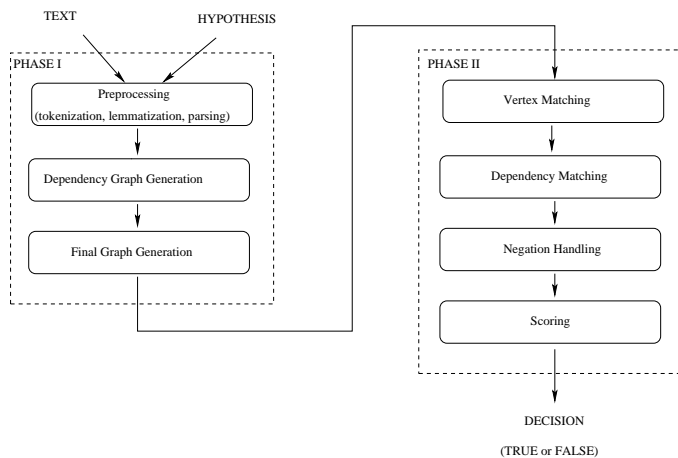


Figure 1: Processing flow in our approach.

Hypothesis and it is accounted for in the score after the entailment decision for the Text-Hypothesis pair without negation is made. If one of the text fragments is negated the decision is reversed while if both are negated the decision is retained (double-negation), and so forth. For example, the Text *Yahoo bought Overture* does not entail Hypothesis *Yahoo did not buy Overture* because even though Text subsumes the Hypothesis without negation (Yahoo did buy Overture), the presence of negation reverses that decision. In Equation 1 the term $\#neg_rel$ represents the number of negation relations between T and H. We recognize our algorithm for negation handling has its limitations and our plans are to further improve in the future.

The entailment approach between a Text and Hypothesis generates a score that indicates the degree of the Hypothesis being subsumed by the Text. In RTE challenges (Dagan, Glickman, & Magnini 2004 2005), the Text is longer, word-wise, than the Hypothesis which is typical for entailment relations among two sentence pairs. However, there are other sentence-to-sentence relations, such as elaboration (McCarthy *et al.* 2007), in which the original sentence, i.e. Text, is shorter than its elaborated counterpart, i.e. Hypothesis. In an elaboration identification task, the challenge would be to decide whether the Hypothesis is an elaboration of the Text. In such cases, our entailment approach can still be applied in a reverse manner: because the Hypothesis is longer than the Text in an elaboration relation then we must check whether the Hypothesis graph subsumes the Text graph. We could simply compute the score $entscore(H, T)$ where the roles of the Text and Hypothesis in the regular entailment case are reversed. We call the regular entailment approach the *Forward* approach and the elaboration approach the *Reverse* entailment approach.

Bidirectional Approach

The approach to entailment can be extended to handle paraphrases. The idea is based on the observation that text A is a paraphrase of text B if and only if text A entails text B and text B entails text A. Thus, a paraphrase score that is the av-

erage of $entscore(A, B)$ and $entscore(B, A)$ can be defined.

$$paraph(A, B) = \frac{entscore(A, B) + entscore(B, A)}{2} \quad (2)$$

Summary of Results

We present in this section the details of the experiments we conducted to evaluate the performance of the newly proposed approach to paraphrase identification. We used the MS Paraphrase Corpus, a standard corpus for paraphrase identification (Dolan, Quirk, & Brockett 2004). MS Paraphrase Corpus (Dolan, Quirk, & Brockett 2004) contains 5801 pairs of sentences collected from various news sources on the web. Each pair is accompanied by "a judgment reflecting whether multiple human annotators considered the two sentences to be close enough in meaning to be considered close paraphrases". The corpus is divided into two subsets: *training* and *test*. There are 2753 TRUE and 1323 FALSE paraphrase instances in the training subset. The test data subset contains 1147 TRUE instances and 578 FALSE instances. For evaluation purposes, we also generated balanced data sets of 1000-1000 (TRUE-FALSE split) and 500-500 instances for training and testing purposes, respectively. The results reported are on the test data sets while the system development/tuning was done on the training set.

The evaluation is automatic and follows the guidelines from RTE (Dagan, Glickman, & Magnini 2004 2005). The judgments (classifications) returned by the system are compared to those manually assigned by the human annotators (i.e. the gold standard). The percentage of matching judgments provides the *accuracy* of the run, i.e. the fraction of correct responses.

As a second measure, a Confidence-Weighted Score [CWS, also known as average precision; (Dagan, Glickman, & Magnini 2004 2005)] is computed. Judgments of the test examples are sorted by their confidence (in decreasing order

Table 1: Performance and comparison of different approaches on MS Paraphrase Corpus [* - result from (Mihalcea, Corley, & Strapparava 2006)].

System	CWS	Accuracy	Precision	Recall	F-measure
Uniform baseline	0.6737	0.6649	0.6649	1.0000	0.7987
Random baseline*	-	0.5130	0.6830	0.5000	0.5780
Forward	0.7852	0.6788	0.7502	0.7751	0.7624
Average	0.8068	0.7061	0.7207	0.9111	0.8048
Reverse	0.7794	0.6736	0.7517	0.7602	0.7560
Kozareva-SVM	-	0.6986	0.9346	0.7066	0.8048
Mihalcea-PMI-IR	-	0.6990	0.702	0.9520	0.8100
Mihalcea-L&C	-	0.6950	0.7240	0.8700	0.7900
Mihalcea-combined	-	0.7030	0.6960	0.9770	0.8130
Qiu	-	0.7200	0.7250	0.9340	0.8160

Table 2: Performance and comparison of different approaches on the balanced corpus derived from MS Paraphrase Corpus.

System	CWS	Accuracy	Precision	Recall	F-measure
Uniform baseline	0.4936	0.5000	0.5000	0.1000	0.6667
Random baseline	0.5112	0.4880	0.4880	0.4900	0.4890
Forward	0.6700	0.6350	0.6060	0.7720	0.6790
Average	0.7046	0.6590	0.6470	0.7000	0.6724
Reverse	0.6732	0.6380	0.6092	0.7700	0.6802

from the most certain to the least certain), calculating the following measure:

$$\frac{1}{n} * \sum_{i=1}^n \frac{\# - correct - up - to - pair - i}{i} \quad (3)$$

where n is the number of the pairs in the test set, and i ranges over the pairs.

The Confidence-Weighted Score varies from 0 (no correct judgments at all) to 1 (perfect score), and rewards the system’s ability to assign a higher confidence score to the correct judgments.

We used the development data to estimate the parameters of the score equation and then applied the equation with the best found parameters to test data. We used linear regression to estimate the values of the parameters and also experimented with balanced weighting ($\alpha = \beta = 0.5, \gamma = 0$). The balanced weighted scheme provides better results. The performance figures reported below are obtained with this balanced scheme. The score provided by the formula in Equation 1 is further used to find the paraphrase decision (TRUE or FALSE) and the level of confidence. Depending on the value of the overall score different levels of confidence are assigned. For instance, an overall score of 0 leads to FALSE paraphrase with maximum confidence of 1. In summary, we obtained an accuracy of 0.7061 and a CWS score of 0.8068 on the standard data set (see *Average* row in Table 1). The results produced by our approach on test data are significant at 0.01 level when compared to the uniform baseline of always guessing the most frequent class (the TRUE class in the MS Paraphrase Corpus) and to the *random baseline* of randomly guessing TRUE or FALSE with equal probability. Our high CWS score indicates that our system is a *confident* paraphrase identifier. This argument becomes stronger when one

thinks of the limited array of resources we use: lexical information enhanced with synonymy from WordNet, syntactic information in the form of dependencies, and antonymy information from WordNet to handle negation. By comparison, other approaches use deeper representations, which are more expensive to build and use, and heavier resources to make paraphrase and entailment decisions (Dagan, Glickman, & Magnini 2004 2005).

Discussion

The results obtained for the original MS paraphrase corpus and for the balanced data set are encouraging but a closer analysis reveals that while the Average system does offer better accuracy and recall it is less powerful than both Forward and Reverse system in terms of precision. Because recall is better for the Average system than the Forward and Reverse systems, the Average system can detect paraphrases very well. However, because precision is not as great, although good, we conclude that the Average system introduces slightly more false positives than the Forward and Reverse systems. For the balanced data set, accuracy of the Average system is still best but interestingly enough the precision and recall pattern changes, i.e. the Average system has better precision but slightly worse recall than the Forward and Reverse systems. In this case, the Average system can less successfully detect paraphrases but when it does so, it does it with high precision. This apparent discrepancy between the behavior of the approaches on different data sets may be explained by the distributions of TRUE and FALSE positive cases in the two data sets. In the original MS paraphrase corpus, the number of positive instance greatly outnumbers the negative instances.

Further Work

There are two main ideas that we plan to explore in the near future. First, we would like to weight words by their specificity when computing the lexical overlap between pairs of sentences. The specificity can be measured using inverted document frequency (IDF) of words derived from large collections of documents, in the order of millions of entries, such as British National Corpus, Text REtrieval Conference corpus (TREC; <http://trec.nist.gov>), or even Wikipedia. The second idea is to learn syntactic patterns for phrase equivalences similar to the patterns discussed in (Lin & Pantel 2001).

Conclusions

The paper presents a software system that implements a new lexico-syntactic approach to the task of paraphrase identification. The approach offers competitive results with other approaches on a standardized data set as shown in Table 1. On the balanced data set (Table 2), the obtained results are significantly better than the uniform or random baselines. Our robust and competitive solution to the task of Paraphrase identification is important to many real world applications such as Natural Language Generation, Question Answering, and Intelligent Tutoring Systems.

References

- Barzilay, R., and McKeown, K. 2001. Extracting paraphrases from a parallel corpus. In *39th Annual Meeting of the Association for Computational Linguistics*, 50–57.
- Dagan, I., and Glickman, O. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of Learning Methods for Text Understanding and Mining*.
- Dagan, I.; Glickman, O.; and Magnini, B. 2004-2005. Recognizing textual entailment. In <http://www.pascal-network.org/Challenges/RTE>.
- Dagan, I.; Glickman, O.; and Magnini, B. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the Recognizing Textual Entailment Challenge Workshop*.
- Dolan, W. B.; Quirk, C.; and Brockett, C. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING 2004*.
- Graesser, A. C.; Olney, A.; Haynes, B.; and Chipman, P. 2005. *Cognitive Systems: Human Cognitive Models in Systems Design*. Erlbaum, Mahwah, NJ. chapter AutoTutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue.
- Ibrahim, A.; Katz, B.; and Lin, J. 2003. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing (ACL 2003)*.
- Iordanskaja, L.; Kittredge, R.; and Polgere, A. 1991. *Natural Language Generation in Artificial Intelligence and Computational Linguistics*. Kluwer Academic. chapter Lexical selection and paraphrase in a meaning-text generation model.
- Kouylekov, M., and Magnini, B. 2005. Recognizing textual entailment with tree edit distance algorithms. In *Proceedings of the Recognizing Textual Entailment Challenge Workshop*.
- Kozareva, Z., and Montoyo, A. 2006. *Lecture Notes in Artificial Intelligence: Proceedings of the 5th International Conference on Natural Language Processing (FinTAL 2006)*. chapter Paraphrase Identification on the basis of Supervised Machine Learning Techniques.
- Landauer, T. K.; Foltz, P.; and Laham, D. 1998. Introduction to latent semantic analysis. *Discourse Processes* 25.
- Leacock, C., and Chodorow, M. 1998. Combining local context and wordnet sense similarity for word sense identification. In *WordNet: An Electronic Lexical Database*. MIT Press.
- Lin, D., and Pantel, P. 2001. Dirt - discovery of inference rules from text. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01)*, 323–328.
- McCarthy, P.; Rus, V.; Crossley, S.; Bigham, S.; Graesser, A.; and McNamara, D. 2007. Assessing entailment with a corpus of natural language. In *Proceedings of the Florida Artificial Intelligence Research Society International Conference (FLAIRS)*. Menlo Park, CA: AAAI Press.
- McNamara, D. S.; Boonthum, C.; Levinstein, I. B.; and Millis, K. 2007. *Handbook of Latent Semantic Analysis*. Mahwah, NJ: Erlbaum. chapter Evaluating self-explanations in iSTART: comparing word-based and LSA algorithms, 227–241.
- Mihalcea, R.; Corley, C.; and Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence (AAAI 2006)*.
- Miller, G. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.
- Monz, C., and de Rijke, M. 2001. *Light-Weight Entailment Checking for Computational Semantics*. 59–72.
- Pazienza, M.; Pennacchiotti, M.; and Zanzotto, F. 2005. Textual entailment as syntactic graph distance: A rule based and svm based approach. In *Proceedings of the Recognizing Textual Entailment Challenge Workshop*.
- Qiu, L.; Kan, M.; and Chua, T. 2006. Paraphrase recognition via dissimilarity significance classification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, 18–26. Association of Computational Linguistics.
- Rus, V., and Desai, K. 2005. Assigning function tags with a simple model. In *Proceedings of Conference on Intelligent Text Processing and Computational Linguistics (CICLing) 2005*.
- Rus, V.; McCarthy, P.; McNamara, D.; and Graesser, A. 2008. A study of textual entailment. *International Journal of Artificial Intelligence Tools*. in press.