# Recognizing Textual Entailment Using Weighted Dependency Relations

Tanik Saikh[1], Sudip Kumar Naskar[2], and Asif Ekbal[1]
{tanik.srf17, asif}@iitp.ac.in[1], sudip.naskar@cse.jdvu.ac.in[2]

[1] Indian Institute of Technology Patna, India
[2] Jadavpur University, Kolkata, India

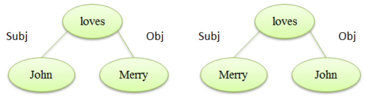
**Abstract.** In this paper, we describe a hybrid approach for Recognizing Textual Entailment (RTE) that makes use of dependency parsing and semantic similarity measures. Dependency triplet matching is performed between dependency parsed *Text* ($T$) and *Hypothesis* ($H$). In case of dependency relation match, we also consider partial matching and semantic similarity between the associated words is calculated with the help of various semantic similarity measures. Importance of various dependency relations with respect to the TE task is computed in terms of their information gain and the dependency relations are weighted accordingly. This paper reports our experiments carried out on the RTE-1, RTE-2 and RTE-3 benchmark datasets using three approaches namely **greedy approach**, **exhaustive search** and **greedy approach with weighted dependency relations**. Experimental results show that weighted dependency relations significantly improve TE performance over the baseline.

**Keywords:** Dependency Parser, Semantic similarity, Textual Entailment, Machine Learning.

## 1 Introduction

An important feature of natural language is the language variability. There are several ways to express a simple matter. A single piece of text can have various meaning or same meaning can be conveyed by different texts. Textual Entailment (TE) is a kind of problem which would be able to capture such situations. It is one of the toughest problems in natural language processing (NLP) which involves rigorous linguistic analysis and machine learning techniques. It is an unidirectional relation [1] between a pair of texts expressions and it exists if a text called *Text (T)* can be logically inferred from the other one called *Hypothesis (H)*. It essentially shows the heredity property between a pair of texts as if $H$ inherits some property from $T$. It is one of the most prominent research topics in the field of NLP with several important applications such as in *Machine Translation [2]*, *Summarization [3]*,*Question Answering [4]*, *Paraphrase Detection[5]* and *Novelty/Plagiarism Detection [6]* in a document/text etc and many more. TE between a pair of texts expressions can be determined by observing the lexical, syntactic, semantic information between that particular pair of texts snippets. Literature shows there are many who proposed various studies on it which include lexical [7], syntactic [8], and semantic [9] information. Recently, there has been much interest in applying deep learning models to RTE [10–14]. These models usually do not perform any linguistic analysis. The proposed method can be expressed as the combination of syntactical divergence and semantic similarity which tries to assign weight to each dependency relations produced by dependency parser. This paper proposed an approach to solving the TE problem by taking the help of dependency parsing information and various semantic similarity measures. Dependency parsing of a particular piece of text essentially provides the syntactic representation of that particular piece of text; it produces a number of triplets, each of which essentially represents a relation between two words, the *Governor (G)* and the *Dependent (D)*, like, the triplet *”nsubj (traded-5, delivery-4)”*, where *G = traded* and *D = delivery.* Various semantic similarity measures were employed for the purpose namely *Wu-Palmer similarity* [15], *Lin similarity* [16] and *path based similarity* [17]. These metrics are generally used to find semantic similarity between a pair of words, phrases or sentences. Lexical matching suffer from a drawback, occasionally it produced a high score for non-textually entailed texts pair, if we take entailment decision between the text pair cited as follows *T: John loves Merry* and *H: Merry loves John*, by considering n-gram matching, it will produce a high score, consequently, the system will mark that pair as entailed, like the unigram and bigram matching between T and H produces 3/3 = 1 and 0/3 = 0 scores respectively. According to unigram matching the sentence pair are textually entails, however they are not.

On the other hand, Dependency Parsing (DP) for the T and H are shown below, where the triplets are *Subj (loves, Merry)* and *Obj (loves, John).* So we can see



DP for T DP for H

none the of *Subj* or *Obj* matches; hence they (text pair) are not entailed, as it is in fact. So dependency matching captures this kind of situations, which is missing in lexical matching. This is the main motivation for using dependency matching in this study.

The motivation behind optimizing dependency relations by assigning weight to them, is the fact that every relation produced by dependency parser for a particular sentence is not equally important, so if we shall be able to assign a weight to each relation and multiply that weight with score obtained from semantic similarity score calculated between two words associated with that particular relation, intuitively, it can be assumed that weighted dependency relation score

will increase our system's performance.
The contributions in this paper can be encapsulated as follows

1. We propound an approach for TE recognition which utilizes dependency parsing information and semantic similarity measures for a T–H pair.
2. We also posit a technique to assign weight to each dependency relation type in order to estimate the importance of each dependency relation with respect to the TE task. We compute the weight of each dependency relation in terms of its normalized information gain.
3. Three sets of experiments namely greedy search, exhaustive search and greedy search with weighted dependency relations are executed.
4. Making use of weighted dependency relations proves to be a very useful technique for recognizing TE.
5. Weighted dependency relations, which we have prepared could be used further for research purpose not only for TE and/or semantic textual similarity but also in any domain of NLP.

### 1.1 Related Works

Since from a decade, researchers have proposed many different approaches to find the TE relation between a pair of texts. These techniques include the use of dependency parsing of texts, employment of various semantic similarity metrics between a pair of texts.

Literature survey shows several studies on RTE using dependency parsing and WordNet-based semantic similarity were performed. Some of the dependency parsing based TE are reported in [18–29]. These models either exploited parsing information from various parsers and/or semantic information from Wordnet. However, to the best of our knowledge, there is no such study which combines dependency parsing (triplet matching technique, with triplet relation optimization) with semantic similarity to RTE.

The first PASCAL RTE challenge [30] provided a standard platform against which systems can be compared on TE scenario and it was essentially the primary attempt to provide a general task that takes into account major semantic inferences across applications. The challenge received a noticeable response from research communities across the globe; 17 participants took part in it. The participating systems obtained relatively low accuracies which suggest that the task is a challenging one and there are various research avenues in this area. In this challenge, the best result was achieved by [31] using the Bilingual Evaluation Understudy (BLEU) algorithm and obtained an accuracy of 70% on RTE-1 datasets Comparable Document (CD) subtask.

A logic-based semantic approach was proposed in [32] which combines the semantic information provided by different resources and extracts new semantic knowledge to improve the systems performance. The tasks of [33, 34] posed machine learning based approaches using conventional similarity metrics (cosine similarity, Jaccard, Dice, etc), along with machine translation (MT) evaluation metrics(BLEU [35] and METEOR [36] [37]) as features on RTE-1 to RTE-3 datasets and Indian Languages(Tamil, Telugu, Hindi, and Punjabi) respectively. The work defined in [38] made use of three MT evaluation metrics (BLEU, METEOR, and TER [39]) and one summary evaluation metric (ROUGE [40]) along with polarity (negation) feature on RTE-1, RTE-2, RTE-3, RTE-4, and RTE-5 and obtained a reasonable output as compared to the best performing results in those tracks. The message was there is a correlation between MT evaluation and TE.

The authors in [41] performed several experiments on RTE-3 datasets. They built a system which solely relies on DIRT (Discovering Inference Rules from Text) [42], which is a collection of paraphrases. The system obtained an accuracy of 59.1% on RTE-3 test set. The work of [43] proposed an unsupervised metric to compute the similarity between word pairs in Web1T data. Their proposed approach made use of dependency tree matching technique between text and hypothesis sentences to compute alignment score based on new similarity metric. These scores further used to predict entailment between T-H pairs. The method yielded an overall accuracy of 50.9% on the RTE4 test set.

## 2 Proposed Approach

T–H pairs are first extracted from the datasets. Parsing of each such T–H pair is performed using the Stanford Dependency parser[3]. The parser produces a set of triplets for each sentence. A similarity matrix of order $m * n$ is created from the dependency parse trees of T and H where m and n are the number of triplets in T and H respectively. It was observed in the TE datasets that the texts are typically longer than the hypotheses and therefore $m > n$ usually. A dependency triplet represents a dependency relation between two words in the sentence, the *Governor (G)* and the *Dependent (D)*. We assign weight to each type of dependency relation based on its information gain on the development set. Each dependency triplet of T is compared against each dependency triplet of H. In case of a full match between a T triplet and an H triplet, a score of 1 is assigned to the corresponding T–H triplet pair and insert that value into the appropriate cell in the matrix. Otherwise, the system looks for a matching relation between the T–H triplet pair. If the relation matches, we consider the semantic similarity between the governing words and the semantic similarity between the dependent words of the two triplets. Finally, the average of these two similarity scores is assigned to that T–H dependency triplet pair. The assumption is that the two governing words or the two dependent words in a T–H triplet pair might not be exactly the same words, but they could be synonymous words or related words. In that case, we should consider assigning that triplet pair some non-zero score. Thus instead of considering only binary scores (i.e., 0 or 1) to a triplet pair, we assign scores between 0 and 1. Three semantic similarity metrics, namely *Wu-Palmer (WUP) similarity* [15], *Lin similarity* [16] and *path-based similarity* [17] have been taken into account for this purpose.

*Wu-Palmer* measures similarity by considering the depth of the two synsets

[3] https://nlp.stanford.edu/software/stanford-dependencies.html

in the WordNet taxonomy, along with the depth of their least common subsumer (LCS). *Lin* measures the similarity between a pair of concepts, say $C_1$ and $C_2$, as $2 * IC(LCS(C_1, C_2))(IC(C_1) + IC(C_2))$ ,where, $IC(x)$ is the information content of $x$. *Path-based metric* determines the semantic similarity between two-word senses as an inverse function of the length of the path linking the two concepts. The average of these three semantic similarity metric's scores is taken into consideration as the partial matching score for a T–H triplet pair and is inserted into the appropriate cell in the matrix. Once the triplet matching process terminates, the matrix populated with scores represents the similarity scores between T–H dependency triplets. Assuming that the rows and columns correspond to the dependency relations in H and T, respectively, two or more nonzero scores in a row of this similarity matrix indicates that the corresponding H triplet matches with multiple T triplets. Similarly, two or more nonzero scores in a column in this similarity matrix suggests that the corresponding T triplet matches with multiple H triplets. However, while computing similarity between a T–H dependency tree pair, any H triplet cannot be allowed to match (or align) with multiple T triplets and vice versa. Thus while computing the dependency triplet alignment between a T–H dependency tree pair, we can consider only a maximum of $min(m, n)$ matching dependency triplets. Therefore, we are interested in finding an optimal combination of $p$ non-zero similarity scores (or cells) from this similarity matrix such that $p < min(m, n)$, every row and column contribute a maximum of 1 similarity score (or cell) in the combination and the sum of the corresponding similarity scores is highest. Three different approaches were adopted to achieve this objective and to generate the final entailment score for the corresponding T–H sentence pair.

### 2.1 Greedy Approach

In this approach, first the maximum of all the non-zero values in the matrix is selected and the rest of the non-zero values in the corresponding row and column are set to zero. In every successive iteration, the same process is repeated and iterations continue until there are no more non-zero values left in the matrix. In case of a tie, we select any of the highest scoring cell. Finally, we take the sum of all the selected elements which is further normalized by the number of triplets present in H to arrive at the semantic similarity entailment score for the corresponding T–H pair.

### 2.2 Exhaustive Search

We exhaustively search over all combinations of $p$ non-zero similarity scores such that $p < min(m, n)$, and every row and column contribute at most one similarity score in any combination and find out the globally best scoring combination (or alignment). It is to be noted, however, that we are not interested in the combination itself; our objective is to find the optimal similarity score for a T-H dependency tree pair and multiple combinations (or alignments) can yield in the same optimal similarity score. Finally, the optimal sum is normalized by *min* *(m, n)* since a maximum of *min (m, n)* similarity scores can contribute to a combination.

### 2.3 Relation Optimization

Ts and Hs are passed through dependency parser, which yields a collection of relations and words associated with those relations (triplets) for a particular sentence. When comparing T–H dependency triplets for finding TE relation between a pair of texts expressions, all the relations (or triplets) are not equally important and there might be some less important relations which contribute less to the TE recognition process than the other relations. However, the proposed method discussed in the previous section assumes uniform weights for all relations. Therefore, in a bid to improve the overall system performance, we try to assign weight to each dependency relation type according to its importance in TE. We compute the relevance of each dependency relation type in TE in terms of its information gain. For every T–H pair in the dataset, we compute the optimal combination of (fully or partially) matching triplets. It is to be noted however that, in this case, we are interested in the optimal combination and not in the optimal similarity score. Subsequently, we find out how many times each particular dependency relation type contributes to an optimal combination. For every relation, for every T–H pair, we check whether the relation contributes to any optimal triplet combination or not; if it does then a value of 1 is assigned to that relation, otherwise, 0 is assigned. For any relation, the 1 values are considered as $child_1$ and 0 values as $child_0$. We calculate the number of $child_1$ and $child_0$ instances for each relation. To calculate the entropy of $child_1$ and $child_0$ of a relation, we also count how many of its $child_1$ and $child_0$ instances belong to the *TRUE* entailment class and *FALSE* entailment class. Finally, the entropy of its $child_i$ is calculated as in Equation 1

$$Entropy_{child_i} = -nt/N_i * log(nt/N_i) - nf/N_i * log(nf/N_i) \tag{1}$$

where, $nt$ is the number of $child_i$ instances in the TRUE class, $nf$ is the number of $child_i$ instances in the FALSE class, and $N_i$ is the number of $child_i$ instances, i.e., $nt$+$nf$. The weighted entropy of each child is computed as in Equation 2.

$$W\_E_{child_i} = -Entropy_{child_i} * N_i/N \tag{2}$$

where, $nC$ is the number of instances of $child_i$, and $N$ is the total number of instances of this particular relation in the whole dataset. We used *add-1* smoothing while calculating the entropy to avoid division by zero and/or zero probability. In effect, 1 is added to each of $nt$ and $nf$, 2 is added to $N_i$ and 4 is added to $N$ while calculating weighted entropy.
Finally, information gain (IG) for the relation is calculated as in Equation 3.

$$IG_{Relation} = 1 - (WE_{Child_1} + WE_{Child_0}) \tag{3}$$

We sort the relations in descending order of their information gain values to assess the importance of the relations and normalize the information gains by the sum of all information gains. These normalized information gain values are considered as the weights for the dependency relations.

## 3 Experiments and Results

This section presents the dataset, experimental setup and the results together with some discussions.

### 3.1 Dataset

There were many international conferences and evaluation tracks have been organized such as at *Pattern Analysis, Statistical Modeling and Computational Learning (PASCAL)*[4], *Text Analysis Conferences (TAC)*[5] organized by the United States National Institute of Standards and Technology (NIST), Evaluation Exercises on Semantic Evaluation (SemEval)[6], National Institute of Informatics Test Collection for Information Retrieval System (NTCIR) [7] since from the year of 2005. These were all dedicated to either recognizing TE between a pair of texts snippets or finding semantic similarity between a pair of texts etc. The experiments were carried out on the datasets released in the PASCAL organized shared task for recognizing textual entailment (RTE) organized in RTE-1, RTE-2 and RTE-3. This paper particularly focusing on two class RTE problem, for that we particularly use RTEs datasets where T–H pairs are defined as two class problem. We would like to port the system to three class problem on Stanford Natural Language Inference (SNLI) Corpus [44] in future. RTE-1 contains 567 and 800 T–H pairs in the development set and test set respectively. Both the development set and test set of RTE-2 and RTE-3 contain 800 entries. Since this work focuses on binary class classification for TE, only RTE-1, RTE-2, and RTE-3 datasets are taken into account. The table 1 shows true vs false entailment statistics in the dataset.

**Table 1.** True-False Entailment pair statistics in the dataset

| | # of T–H pairs with class | |
|---|---|---|
| Dataset | True | False |
| RTE-1 | 283 | 284 |
| RTE-2 | 400 | 400 |
| RTE-3 | 412 | 288 |

### 3.2 Setup

We set various threshold values for taking the entailment decision between T–H pairs and find out the optimum threshold value which maximizes system performance on the development set. The threshold value is obtained by following a

[4] http://pascallin.ecs.soton.ac.uk/Challenges/

[5] http://www.nist.gov/tac/tracks/index.html

[6] http://semeval2.fbk.eu/semeval2.php

[7] http://research.nii.ac.jp/ntcir/ntcir-9/

hill-climbing approach where steps are chosen which take us to the highest peak. Following this methodology, if the result obtained with a particular threshold value is better than with another threshold value, we consider our next threshold around the one that yields a better result. Finally, the threshold that provides the best performance on the development set is applied on the corresponding test set. We ran three sets of experiments for each dataset using greedy approach, exhaustive search and greedy search with weighted dependency relations.

### 3.3 Results

The results obtained with the three approaches on the three datasets are presented in Table 2, where Greedy search, Exhaustive search and greedy search with dependency relations, optimization approaches are represented as Greedy, Exhaustive and RO respectively. A general trend can be observed from the results. The baseline greedy approach produces TE accuracy in the range 55%–56% on all the three datasets. The exhaustive search, as expected, results in some (about 1%–2%) improvements over the baseline greedy approach and produces TE accuracy in the range 56%–58%. The relation optimization approach produces significant improvements (about 5%–8%) over exhaustive search and yields TE accuracy in the range 62%–66%. The improvements provided by an exhaustive search over greedy search and relation optimization approach over exhaustive search are systematic. The relation optimization approach produces the best performance on all three test sets and exhibits 63.12%, 62.37%, and 66.51% accuracies on RTE1, RTE2, and RTE3 respectively. Table 2 also reports the best performances reported in the literature on these three datasets, which are 70% [31], 75% [45] and and 80% [46] on RTE1, RTE2 and RTE3 respectively. Thus the proposed approach falls short of the state-of-the-art results obtained on these tracks. However, the proposed approach is a simple unsupervised approach solely relying on matching dependency trees and it does not rely on any other sophisticated tools or techniques, whereas the works reporting state-of-the-art results used many different techniques, e.g., [31] made use of word overlapping method of the BLEU algorithm [35]. [45] used lexical relations, N-gram subsequence, syntactic matching, semantic role labeling, Web-Based statistics etc., [46] considered discourse commitments, lexical alignment, knowledge extraction

**Table 2.** Evaluation results on the TE task

| Datasets | Approach | Threshold | Accuracy(%) | Best |
|---|---|---|---|---|
| RTE-1 | Greedy | 0.4 | 55.75 | 70 |
| | Exhaustive | 0.0875 | 56.62 | |
| | RO | 0.4625 | 63.12 | |
| RTE-2 | Greedy | 0.375 | 56.25 | 75 |
| | Exhaustive | 0.125 | 58.56 | |
| | RO | 0.25 | 62.37 | |
| RTE-3 | Greedy | 0.275 | 55.75 | 80 |
| | Exhaustive | 0.15 | 58.25 | |
| | RO | 0.55 | 66.51 | |

from various knowledge Bases, etc. We also compare the results obtain by the proposed system with some existing works exploiting dependency information to TE. Table 3 shows some results of other's system and the results of the proposed system. The proposed system outperforms the existing system.

**Table 3.** Comparison with some systems

| Group | Dataset | Accuracy(%) |
|---|---|---|
| Proposed Approach | RTE-1 | 63.12 |
| | RTE-2 | 62.37 |
| | RTE-3 | 66.51 |
| [41] | RTE-3 | 59.1 |
| [42] | RTE-4 | 50.9 |
| [18] | PETE | 73.75 |
| [47] | RTE-4 | 53.86 |

### 3.4 Error Analysis

We manually analyzed some of the erroneous cases and the observations are given below.

1. Stanford dependency parser sometimes produces erroneous results, even for short sentences. We also noticed that the Stanford phrase structure parser is excellent compared to the Stanford dependency parser for such cases. In future, we would like to incorporate that into the existing framework to make a comparative study.
2. For many related word pairs (like *Cease and network*, *ended and went*, *Police and Police*, *forms and document*, *oil and prices*, *traded and rose* and many more) all the WordNet-based metrics (Wu-Palmer, Lin, path-based similarity) produce semantic similarity score 0, which affects the entailment scores, and in turns affects the entailment decision.
3. It was observed from the dataset that there are many instances of *FALSE* entailment T–H pairs in the training set that yield TE score of 0.5 or higher which makes the threshold, and hence the TE recognition process, difficult to learn. Table 4 presents statistics of such T–H pairs in the datasets.

**Table 4.** Statistics of high scoring (≥0.5) *FALSE* TE entailment pairs in the Development sets

| Datasets | # of T–H pairs | | |
|---|---|---|---|
| | Greedy | Exhaustive | RO |
| RTE-1 | 86 | 76 | 76 |
| RTE-2 | 105 | 72 | 167 |
| RTE-3 | 81 | 58 | 108 |

4. It was also observed that high scoring (≥0.5) *FALSE* TE entailment pairs typically contain lots of *Named Entities (NE)* in both T and H. This needs further investigation.

## 4 Conclusion and Future Work

The paper presents a hybrid approach for TE recognition which exploits dependency parsing information and semantic similarity measures for a T–H text pair. We also tendered a technique which is based on information gain to assign weight to each dependency relation type. We carried out 3 sets of experiments - baseline greedy, exhaustive search and relation optimization, on RTE 1–3. The thresholds were learned from the development sets using a hill-climbing approach. We successfully demonstrated our hypothesis that all the dependency relations are not equally important for the task of TE recognition. Finding the importance of the dependency relations for the TE task through information gain and using them as weights in the T–H dependency tree similarity calculation is the major contribution in the proposed work, which resulted in significant improvements in the TE recognition task.

In future, we would like to apply the proposed system for three class TE problem like what is defined in RTE-4, RTE-5, Stanford Natural Language Inference (SNLI) Corpus [44] and recently released Multi Genre Natural Language Inference corpus (MultiNLI) proposed by [48]. We are also planning to employ Word2Vec model based distributional semantic similarity to remedy the problem of WordNet based semantic similarity.

## References

1. Tatar, D., Serban, G., Mihis, A.D., Mihalcea, R., et al.: Textual Entailment as a Directional Relation. Journal of Research and Practice in Information Technology **41**(1) (2009) 53
2. Hutchins, W.J., Somers, H.L.: An Introduction to Machine Translation. London: Academic Press (1992)
3. Radev, D.R., McKeown, K.R.: Generating natural language summaries from multiple on-line sources. Comput. Linguist. **24**(3) (sep 1998) 470–500
4. Green, Jr., B.F., Wolf, A.K., Chomsky, C., Laughery, K.: Baseball: An Automatic Question-Answerer. In: Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference. IRE-AIEE-ACM '61 (Western), New York, NY, USA, ACM (1961) 219–224
5. Socher, R., Huang, E.H., Pennington, J., Ng, A.Y., Manning, C.D.: Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In: Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain. (2011) 801–809
6. Lambert, P., Blanchard, J., Guillet, F., Kuntz, P., Suignard, P.: Novelty Detection in Text Streams - A Survey. In: European Conference on Data Analysis. Proc. of the European Conference on Data Analysis, Luxembourg, Luxembourg (2013)
7. Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge. In: Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment. MLCW'05, Berlin, Heidelberg, Springer-Verlag (2006) 177–190

8. Vanderwende, L., Dolan, W.B.: What Syntax Can Contribute in the Entailment Task. In: MLCW. Volume 3944 of Lecture Notes in Computer Science., Springer (2005) 205–216
9. Burchardt, A., Reiter, N., Thater, S., Frank, A.: A Semantic Approach to Textual Entailment: System Evaluation and Task Analysis. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. RTE '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 10–15
10. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: Learning natural language inference from a large annotated corpus. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, Association for Computational Linguistics (2015) 632–642
11. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kocisky, T., Blunsom, P.: Reasoning about entailment with neural attention. In: International Conference on Learning Representations (ICLR). (2016)
12. Wang, S., Jiang, J.: Learning natural language inference with lstm. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Association for Computational Linguistics (June 2016) 1442–1451
13. Parikh, A., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, Association for Computational Linguistics (November 2016) 2249–2255
14. Sha, L., Chang, B., Sui, Z., Li, S.: Reading and thinking: Re-read lstm unit for textual entailment recognition. In: COLING. (2016)
15. Wu, Z., Palmer, M.: Verbs Semantics and Lexical Selection. In: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics. ACL '94, Stroudsburg, PA, USA, Association for Computational Linguistics (1994) 133–138
16. Lin, D.: An Information-Theoretic Definition of Similarity. In: Proceedings of the Fifteenth International Conference on Machine Learning. ICML '98, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1998) 296–304
17. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet::Similarity: Measuring the Relatedness of Concepts. In: Demonstration Papers at HLT-NAACL 2004. HLT-NAACL–Demonstrations '04, Stroudsburg, PA, USA, Association for Computational Linguistics (2004) 38–41
18. Basak, R., Naskar, S.K., Pakray, P., Gelbukh, A.F.: Recognizing Textual Entailment by Soft Dependency Tree Matching. Computacin y Sistemas **19**(4) (2015)
19. Herrera, J., Peñas, A., Verdejo, F.: Textual Entailment Recognition Based on Dependency Analysis and *WordNet*. In: Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers. (2005) 231–239
20. Snow, R., Vanderwende, L., Menezes, A.: Effectively using syntax for recognizing false entailment. In: Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, New York City, USA, Association for Computational Linguistics (June 2006) 33–40
21. Pakray, P., Bandyopadhyay, S., Gelbukh, A.: Dependency parser based textual entailment system. In: Artificial Intelligence and Computational Intelligence, International Conference on(AICI). Volume 01. (10 2010) 393–397
22. Rus, V., G.A.M.P..L.K.: A study on textual entailment. In: Proc. of 17th International Conference on Tools with Artificial Intelligence (ICTAI05). IEEE,. (2005) 326–333

23. Marsi, E., Krahmer, E., Bosma, W., Theune, M. Number 2. In: Normalized Alignment of Dependency Trees for Detecting Textual Entailment. Springer Verlag (4 2006) 56–61
24. Kouylekov, M., Magnini, B.: Recognizing textual entailment with tree edit distance algorithms. In: PASCAL Challenges on RTE. (2005) 17–20
25. Haghighi, A.D., Ng, A.Y., Manning, C.D.: Robust textual inference via graph matching. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 387–394
26. Sidorov, G.: Should syntactic n-grams contain names of syntactic relations? Int. J. Comput. Linguistics Appl. **5**(2) (2014) 25–47
27. Dash, S.K., Pakray, P., Gelbukh, A.F.: Learning Physiotherapy through Virtual Action. Computación y Sistemas **20**(3) (2016) 477–482
28. Alabbas, M., Ramsay, A.: Dependency tree matching with extended tree edit distance with subtrees for textual entailment. In: Federated Conference on Computer Science and Information Systems - FedCSIS 2012, Wroclaw, Poland, 9-12 September 2012, Proceedings. (2012) 11–18
29. Marsi, E., Krahmer, E., Bosma, W., Theune, M.: Normalized Alignment of Dependency Trees for Detecting Textual Entailment. In Magnini, B., Dagan, I., eds.: Second PASCAL Recognising Textual Entailment Challenge. (April 2006) 56–61
30. Dagan, I., Glickman, O.: Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability. In: Learning Methods for Text Understanding and Mining. (jan 2004)
31. Perez, D., Alfonsecaia, E., Rodrguez, P.: Application of the Bleu Algorithm for Recognising Textual Entailments. In: Proceedings of the Recognising Textual Entailment Pascal Challenge. (2005)
32. Tatu, M., Moldovan, D.: A Semantic Approach to Recognizing Textual Entailment. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. HLT '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 371–378
33. Saikh T., Naskar S.K., G.C.B.S.: Textual Entailment using Different Similarity Metrics. In: In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2015. Lecture Notes in Computer Science,vol 9041. Springer, Cham. (2015)
34. Saikh, T., Naskar, S.K., Bandyopadhyay, S.: JU_NLP@DPIL-FIRE2016: Paraphrase Detection in Indian Languages - A Machine Learning Approach. In: Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016. (2016) 275–278
35. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Stroudsburg, PA, USA, Association for Computational Linguistics (2002) 311–318
36. Banerjee, S., Lavie, A.: METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association of Computational Linguistics (ACL-2005), Ann Arbor, Michigan, June 2005. (2005)
37. Lavie, A., Agarwal, A.: METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In: Proceedings of the Second Workshop on Statistical Machine Translation. StatMT '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 228–231

38. Saikh, T., Naskar, S., Ekbal, A., Bandyopadhyay, S.: Textual Entailment using Machine Translation Evaluation Metrics. In: Computational Linguistics and Intelligent Text Processing - 18th International Conference on Computational Linguistics and Intelligent Text Processing, CICLing 2017, Budapest, Hungary, April 17 to 23, 2017. (2017)
39. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A Study of Translation Edit Rate with Targeted Human Annotation. In: In Proceedings of Association for Machine Translation in the Americas. (2006) 223–231
40. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26, 2004. (2004)
41. Marsi, E., Krahmer, E., Bosma, W.: Dependency-based Paraphrasing for Recognizing Textual Entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. RTE '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 83–88
42. Lin, D., Pantel, P.: DIRT @SBT@Discovery of Inference Rules from Text. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '01, New York, NY, USA, ACM (2001) 323–328
43. Yatbaz, M.A.: RTE4: Normalized Dependency Tree Alignment Using Unsupervised N-gram Word Similarity Score. In: Proceedings of the First Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008, Gaithersburg, Maryland, USA, NIST (2008)
44. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A Large Annotated Corpus for Learning Natural Language Inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015. (2015) 632–642
45. Hickl, A., Bensley, J., Williams, J., Roberts, K., Rink, B., Shi, Y.: Recognizing Textual Entailment with lccs Groundhog System. (2005)
46. Hickl, A., Bensley, J.: A Discourse Commitment-based Framework for Recognizing Textual Entailment. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. RTE '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 171–176
47. Pakray P., Gelbukh A., B.S.: A Syntactic Textual Entailment System based on Dependency Parser. In: In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2010. Lecture Notes in Computer Science. Volume 6008., Berlin, Heidelberg, Springer (2010)
48. Williams, A., Nangia, N., Bowman, S.R.: A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. CoRR **abs/1704.05426** (2017)