Semi-supervised Textual Entailment on Indonesian Wikipedia Data

Ken Nabila Setya and Rahmad Mahendra

Faculty of Computer Science, Universitas Indonesia
Depok, 16424 West Java, Indonesia
ken.nabila@ui.ac.id, rahmad.mahendra@cs.ui.ac.id

Abstract. Recognizing Textual Entailment (RTE) is a research in Natural Language Processing that aims to identify whether there is an entailment relation between two texts. Textual Entailment has been studied in a variety of languages, but it is rare for the Indonesian language. The purpose of the work presented in this paper is to conduct the RTE experiment on Indonesian language dataset. Since manual data creation is costly and time consuming, we choose semi-supervised machine learning approach. We apply co-training algorithm to enlarge small amounts of annotated data, called seeds. With this method, the human effort only needed to annotate the seeds. The data resource used is all from Wikipedia and the entailment pairs are extracted from its revision history. Evaluation of 1,857 sentence pairs labelled with entailment information using our approach achieve accuracy 76%.

Keywords: Textual Entailment, Co-training, Wikipedia revision history, Indonesian language, semi-supervised, LSTM

1 Introduction

An entailment relation holds between two textual fragments (i.e. text T and hypothesis H) when the meaning of H can be inferred from the meaning of T [9]. Textual entailment recognition is useful to support other Natural Language Processing (NLP) tasks, such as Information Extraction, Question Answering [23], Machine Translation [24], Information Retrieval [8], and Summarization [18].

Not only in English, Recognizing Textual Entailment (RTE) tasks have been also conducted in some other languages, such as Italian [5], Spanish [25], German [31], Arabic [1], and Greek [21]. To our knowledge, no prominent work on RTE task for Indonesian language.

In the data-driven era, textual entailment relation can be identified by machine learning approaches. To obtain good result, the classifier need to learn from more examples in training data. Hickl et al. [14] automatically constructed datasets and successfully improve the performance of textual entailment engines by up to ten percent. The availability of very large scale annotated corpus has stimulated research on natural language inference to achieve accuracy score more than 77% [6].

This paper discusses our work on applying semi-supervised method to address the lack of labeled RTE data for Indonesian language. The paper is organized as follows. Section 2 provides information about previous work on creating Textual Entailment data as well as machine learning approach for Textual Entailment recognition. We discuss our proposed methodology in Section 3. We report the experiment result in Section 4 and conclude in Section 5.

2 Related Work

A spectrum of approaches has been proposed for RTE task [2]. Logic-based approach transforms natural language expression into logical meaning representation and then makes reasoning by invoking theorem prover [4] [27]. While, machine learning approach learns the set of features from the examples to recognize entailment relationship between pair of text [11,15,16,19,30]. Using maching learning, entailment decision problem can be considered as a classification problem.

Various methods work on different level of language representation, such as surface string, syntactic, or semantic representation. Several measurements can be combined to characterize entailment relation, e.g. lexical overlapping and longest common sub-sequence between surface representation of T and H, tree edit distance of dependency tree representation of input expression, and semantic distance between word meaning across text fragment pair. When applying machine learning approach, such distances or similarity measures can be leveraged as hand-crafted features. In the recent years, deep neural networks successfully deliver remarkable result to the NLP tasks, including RTE. LSTM model has outperformed similarity-based or traditional machine learning-based classifiers to tackle the RTE task [6, 26, 28].

Experimenting with textual entailment recognizer requires the datasets containing both positive and negative input pairs. Commonly used data for the evaluation of textual entailment and natural language inference task for English language are dataset from The PASCAL Recognising Textual Entailment Challenge [10, 12, 13]. Larger benchmark is the data for SemEval 2014 shared task, called Sentences Involving Compositional Knowledge (SICK) that consists of 10,000 sentence pairs; each annotated for relatedness in meaning [20]. Bowman [6] issues the Stanford Natural Language Inference (SNLI) corpus, a collection of 570K sentence pairs labeled for entailment, contradiction, and semantic independence.

While the annotation effort of RTE data is done by the experts, the annotation of SICK and SNLI dataset relies much on manual labor via crowdsourcing system. Apart from human intervention in collecting and labeling data, several works attempt to acquire data (semi-)automatically. Burger generates an entailment recognition corpus from the news headline and the first paragraph of a news article. However, this approach is limited to grab the positive entailment pairs only [7]. Hickl improved upon this work by including negative examples using heuristic rules (e.g., sentences connected by `although`, `otherwise`, and `but`) [14].

Zanzotto conducted the experiments using the co-training method on Wikipedia data to expand the Textual Entailment corpus [29]. Wikipedia revision history can provide two views that are required for the application of Co-training methods. The two views used in their research are the pairs of texts before and after revision and the author comments when revising the article page. The first view of data is represented in syntactic features to be classified using the SVM-light classifier. Whereas, the second view is represented by the bag-of-word.

Our proposed method to approach RTE problem for Indonesian language is semi-supervised. To expand the data from small amount of annotated data, we adopt co-training algorithm from Zanzotto work with slight modification. Instead of using syntactic parse tree (which has not been extensively and carefully studied in Indonesian language) to engineer the features for the classifier, we play with embedding vector representation of words in text pairs that are trained in LSTM model.

3 Methodology

We utilize Indonesian Wikipedia as the source of Textual Entailment dataset. Wikipedia all articles and revision history data in XML format are downloaded from Wikimedia dump site[1]. The text contents are obtained after cleaning Wiki markup language format from Wikipedia all articles using Wikipedia Extractor tool [2]. The texts are segmented into list of sentences by rule-based sentence splitter. A collection of sentences is pre-trained to construct word embedding model using the word2vec [22]. On the other hand, Wikipedia's revision history is taken to produce candidate of entailment pairs. The whole picture of the methodology of this study can be seen in the Figure 1.

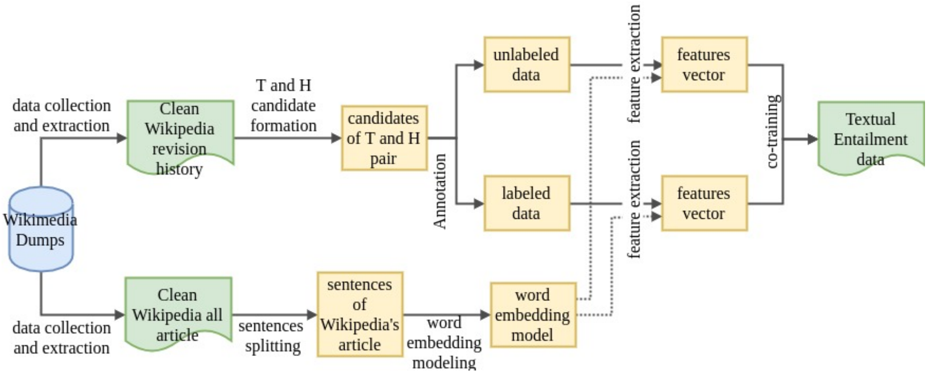


Fig. 1. Proposed Methodology

[1] https://dumps.wikimedia.org/

[2] https://github.com/attardi/wikiextractor

3.1 T and H Candidate Generation

Wikipedia's revision history stores all changes made to any article page. Each revision has its own identifier and the parent identifier (the identifier of previous version of the article). We collect pairs of texts before and after revision for each revision. For each pair, we also extract the comment post that the author wrote when revising Wikipedia article page.

Preprocessing Vandalism Edits Since Wikipedia is written collaboratively, the content quality may be vulnerable. Whereas some revisions of Wikipedia pages are for constructive purpose (i.e. either inserting new information to the text or deleting old information), several edits are vandalisms that deliberately compromise Wikipedia's integrity. At least, vandalism edits are characterized in two ways: (1) the occurence of vulgar and abusive languages, interjections to scold or to make a joke, and/or emoticons. For example: sentence *Chairil Anwar adalah seorang penulis puisi hahahaha* (en: Chairil Anwar is a poet hahahaha) and (2) falsifying the information by changing the meaning. For example: sentence *Andi Hermanto ganteng adalah salah satu mantan presiden Indonesia* (en: The handsome Andi Hermanto was an ex-president of Indonesia). In fact, a man named Andi Hermanto has never be a president.

In the context of our task goal, the first type of vandalism edit needs to be pre-processed. A dictionary of abusive words is compiled. When any word in this dictionary is found in the Wikipedia text revision, it is replaced by the empty string. On the other hand, the text fragments belonging to the second type of vandalism edit is retained as it is. The latter type can be potential candidate of negative entailment pair.

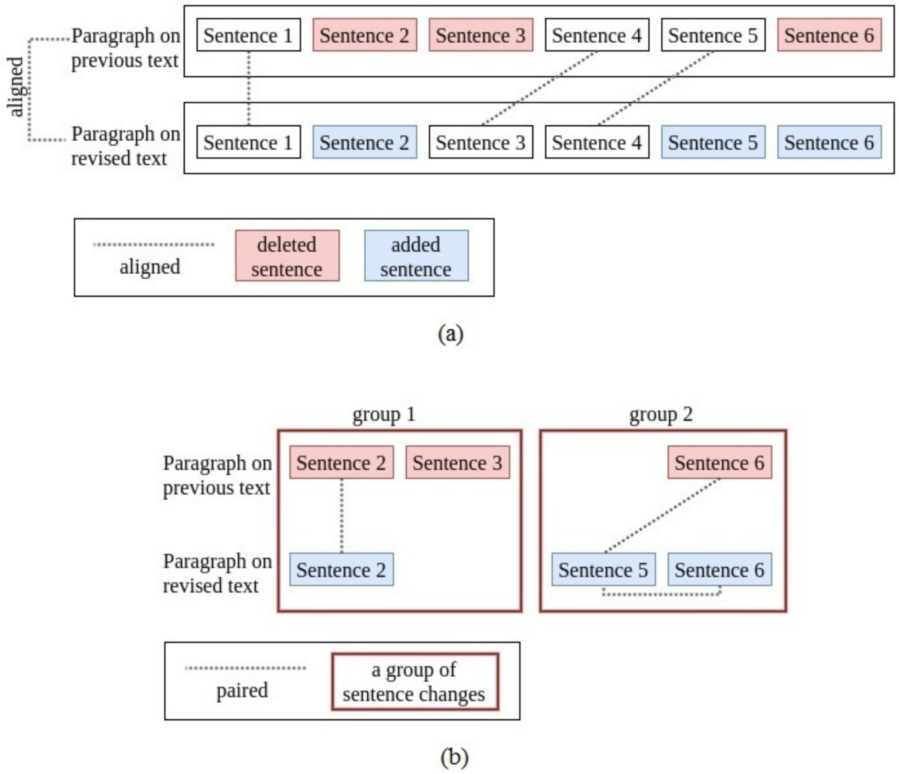
Sentence Alignment and Pairing For a particular revision pair, suppose that the text before revision consists of m sentences $(b_1, b_2, ..., b_m)$ and the text after revision consists of n sentences $(a_1, a_2, ..., a_n)$. We can align two sentences b_i $(1 \leq i \leq m)$ and a_j $(1 \leq j \leq n)$ if both are identical or differentiable by spelling variation. We apply edit distance algorithm to detect typo errors.

The sentence b_i that does not have any alignment is the sentence that previously exists and then deleted after the revision process. While, the sentence a_j that is not aligned with any sentence is introduced by author when revising the article page. Those sentences without alignment are considered as component of entailment pair following three assumptions below.

1. The sentence deleted from the text before revision are paired with newly-added sentences in the text after revision. We infer that new sentence substitutes information of deleted sentence.
2. If the number of deleted sentences (i.e. m) is more than the number of added sentences (i.e. n, $m > n$), then only first n deleted sentences are aligned with added sentences after revision (b_1 is paired with a_1, b_2 with a_2, ..., b_n with a_n). We think that the rest deleted sentences $(b_{n+1}, ..., b_m)$ can be omitted by the author because it is unimportant or other undesirable cases.

3. If the number of added sentences (i.e. n) is more than the number of deleted sentences (i.e. m, $m < n$), then first m added sentences are aligned with deleted sentences before revision (b_1 is paired with a_1, b_2 with a_2, ..., b_m with a_m). In addition, the rest of added sentences are paired with previous sentence (i.e. a_{m+1} is paired with a_m, a_{m+2} with a_{m+1}, ..., a_n with a_{n-1})

Figure 2 illustrates the sentence alignment and pairing process. To determine which sentence is T and which one is H in the entailment relation T → H, we compare the length of sentence before and after revision. The longest one is assigned as T.



(a)

(b)

Fig. 2. Sentences alignment and pairing to obtain T and H pair candidate

3.2 Seed Annotation

Manual annotation is performed on a small portion of the T and H pair candidates. 400 pairs are randomly chosen and each of them is judged by three different annotators. The Kappa agreement [17] of annotation result is 0.827. In the final data that has been validated, we have 223 pairs annotated with E (entailment) label and the rest, 177 pairs, are with NE label (not entailment) label. Annotated data is used as the initial seed for the semi-supervised process.

3.3 Entailment Classification

The co-training [3] method needs two views that are conditionally independent. We assign the candidate pairs of T and H as the first view and author's comment post as the second view.

We implement LSTM model following Bowman architecture [6] as the classifier on the first view of our data. We apply word embedding as the features. Our proposed model architecture is described in Figure 3.

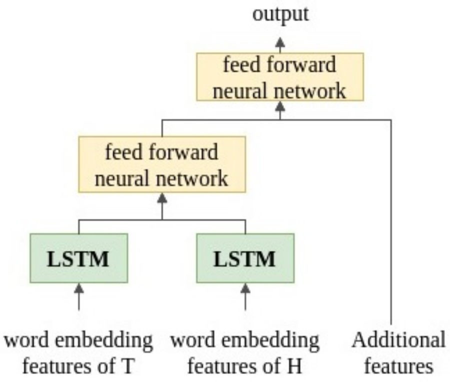

output
feed forward neural network
feed forward neural network
LSTM
LSTM
word embedding features of T
word embedding features of H
Additional features

Fig. 3. Model Architecture of First View

We augment some additional features beside word embedding of T and H sentences to strengthen lexical relationships between T and H. To extract additional features, first we apply words alignment process between T and H pairs. We align identical word occurring in both T and H parts. The rest of words in each part are clustered based on neighboring aligned word. We then align the corresponding cluster between T and H pairs. There is possibility that a particular cluster in one part (T or H) is aligned to empty cluster in another part. The additional features that we use in our model are as follows.

1. The similarity of embedding vector of corresponding non empty clusters between T and H pairs.
2. The number of cluster(s) in T paired with an empty cluster in H, and vice versa.
3. The number of identical words aligned between T and H pairs.
4. The longest sub-sentence of T and H.

5. The boolean value indicating that T and H begin with the same word.

Figure 4 illustrate an example of engineering process of additional features that is used by the classifier on first view.

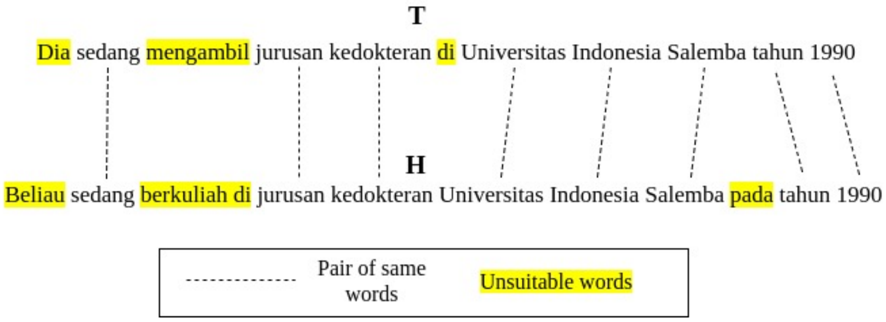

T
Dia sedang mengambil jurusan kedokteran di Universitas Indonesia Salemba tahun 1990
H
Beliau sedang berkuliah di jurusan kedokteran Universitas Indonesia Salemba pada tahun 1990
Pair of same words
Unsuitable words

Fig. 4. Example of words alignment

The aligned words in aforementioned example are *sedang, jurusan, kedokteran, Universitas, Indonesia, Salemba, tahun,* and *1990.* There are 4 unsuitable word clusters: $\{\{dia\}, \{beliau\}\}$, $\{\{mengambil\}, \{berkuliah,\ di\}\}$, $\{\{di\}, \{\}\}$, and $\{\{\}, \{pada\}\}$.

1. The similarity of non-empty clusters of unsuitable word between T and H. The calculation of this feature for the example is

$$\frac{(sim(\{dia\},\{beliau\})+sim(\{mengambil\},\{berkuliah,\ di\})}{2}$$

2. A value in the range 0 to 1 that describes the number of non-empty group(s) in T paired with an empty group in H. To produce values in the range of 0 to 1, the number can be incorporated into the sigmoid function. Since there is only one pair, that is $\{\{\}, \{pada\}\}$, the value of this feature for given example is $sigmoid(1)$
3. A value in the range 0 to 1 that describes the number of non-empty group(s) in H paired with an empty group in T. The value of this feature for given example is $sigmoid(1)$ as only one pair exists, i.e. $\{\{di\}, \{\}\}$
4. A value in the range 0 to 1 that represents number of identical words aligned between T and H pairs. To obtain a value between 0 and 1, the number of words occurring in both T and H should be divided by the total number of words in the text T or H. There are eight pairs of same words on example, so the feature value is

$$(2 \times 8)/(11 + 12) = 0.59$$

5. A value in the range 0 to 1 that represents longest sub-sentence of T and H. The longest sub-sentence in given example is *Universitas Indonesia Salemba*, so the feature value is

$$(2 \times 3)/(11 + 12) = 0.26$$

6. If T and H begin with the same word, the value of the feature is 1, else 0. In the example, T starts with word *dia*, while H starts with word *beliau.* So, the feature value is 0.

Meanwhile, the second classifier runs on the bag-of-words features extracted from author's comment text. We shortlist top frequent phrases that occur on the candidate of T or H pairs. We test several supervised machine learning algorithms on second view of data.

3.4 Co-training

We conduct the experiment on 15,000 candidate pairs of T and H. The co-training algorithm used in our experiment is adjusted from original method introduced by [3]. Our algorithm is presented as follow.

```
Data: L for labeled data, U for unlabeled data
Take k random data from U as U';
set failed iteration 0;
while failed iteration < n do
    Train classifier h1 with first view of L;
    Train classifier h2 with second view of L;
    Classify U' with h1 obtaining U1';
    Classify U' with h2 obtaining U2';
    Assign label for best-classified examples in U1' and U2' and add them to L;
    if there is no example added to L then
        failed iteration++;
    else
        set failed iteration 0;
    end
    Take k data from U to replace the content of current U ';
end
```

Algorithm 1: Co-training

An unlabeled data is tagged a label if both classifiers agree on assigning the label with confidence level more than a specific threshold value (0.9 for first classifier, 0.5 for the second). Co-training ceases when the two classifiers are unable to agree on any of the same labels in n consecutive iterations. Another important parameter is k, a amount of unlabeled data to be retrieved per iteration. After conducting empirical experiment, the values of n and k are respectively set at 3 and 500.

4 Result and Evaluation

Before applying co-training algorithm, we test the performance of the classifiers on the seed data using the 10-fold cross-validation setting. LSTM model as first view classifier scores accuracy 58%. When the hand-crafted features augmented into LSTM, the accuracy of the classifier improves to 69%. After evaluating several traditional supervised machine learning approach, we choose Multinomial Naive Bayes as the classifier to train the model from second view of our data.

In the end, our study yields 1,857 sentence pairs that is iteratively labeled by co-training algorithm. The amount of data being labeled in each iteration is disparate, ranging from 0 to 58. We evaluate the result in 5 iterations (hereafter called the iteration group) by taking a random sample of 13 pairs for each label. The selected data is manually evaluated by human. The average accuracy on label classes E and NE are respectively 82% and 67%. Overall, the accuracy of classification is 76%. Figure 5 shows the sampling accuracy of each iteration group.

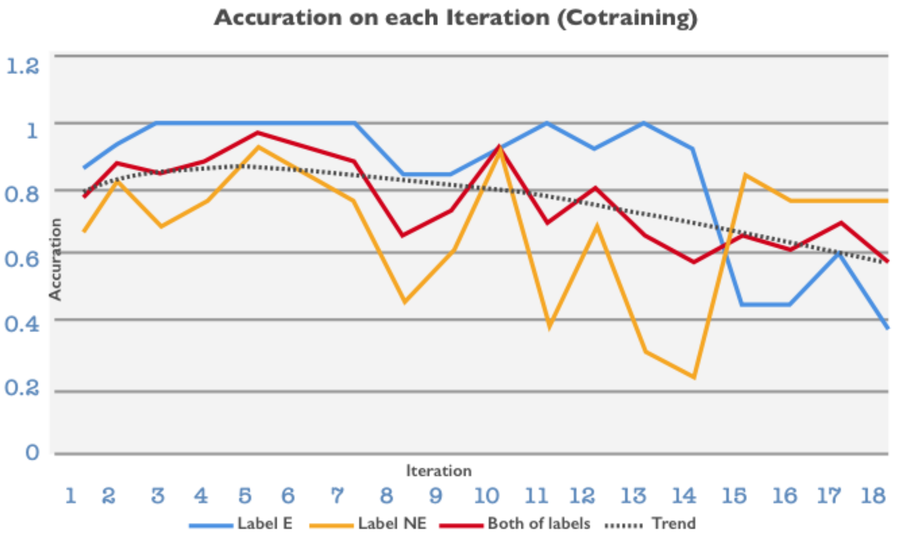


Fig. 5. Accuracy of iteration group on co-training

We observe the several cases in which co-training still make mistakes to assign label, i.e.:

- The difference between T and H is just the use of terms in foreign language (i.e. lexical substitution from English to Indonesian word),
- The difference between T and H is related to quantification problem, such as variation of number writing system (e.g. T uses term *5th*, while H uses

term *fifth*). Another case is determining math inference between two text fragments.

From our experiment, we observe that most revision edits in Indonesian Wikipedia history data are generally paraphrases, or altering particular words with its synonyms. So, the instance of positive entailment relation is quite dominating.

5 Conclusion and Future Works

In this study, we apply a semi-supervised approach, using Co-training algorithm, to build an Indonesian Textual Entailment dataset from the scratch. We obtain data from Wikipedia revision history in Indonesian language through several stages of processing, namely: Wikipedia text extraction, T and H candidate formation, seed annotation, and feature engineering. The method requires data with two independent views. We extract the pairs of text before and after the revision (i.e. a pair of T and H) as first view and the author comment after revised the text as second view. By taking advantage of small portion of labeled data as seeds, co-training method automatically tags unlabeled data using two separate classifiers in each view. Wikipedia revision history is sufficient to produce considerably high-quality large dataset. Starting from only 400 annotated seeds, we can increase the size of the Indonesian Textual Entailment dataset by adding 1,857 new labeled data.

Although the performance of this initial study is good enough for pioneering of Indonesian Textual Entailment research, the further improvements are needed. The exploration on features engineering (especially syntactic and semantic features) is expected to gain the accuracy of classification. More careful inspection can be performed to learn better parameters for co-training algorithm. We also intend to test our methodology on other data sources, i.e. online news or social media.

Acknowledgement

We thank to Mirna Adriani for input and comments. This research was partially supported by PITTA UI Grant Contract No. 410/UN2.R3.1/HKP.05.00/2017. The first author was also partially funded by Bukalapak

References

1. Alabbas, M.: A dataset for arabic textual entailment. In: RANLP. pp. 7–13 (2013)
2. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research 38(1), 135–187 (May 2010)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory. pp. 92–100. Madison, WI (1998)

4. Bos, J., Markert, K.: Recognising textual entailment with logical inference. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (2005)
5. Bos, J., Zanzotto, F.M., Pennacchiotti, M.: Textual entailment at evalita 2009. In: Proceedings of EVALITA 2009. pp. 1–7 (2009)
6. 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). Association for Computational Linguistics (2015)
7. Burger, J., Ferro, L.: Generating an entailment corpus from news headlines. In: Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment. pp. 49–54. EMSEE '05, Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
8. Clinchant, S., Goutte, C., Gaussier, E.: Lexical entailment for information retrieval. In: ECIR. vol. 3936, pp. 217–228. Springer (2006)
9. 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. pp. 177–190. Springer-Verlag, Berlin, Heidelberg (2006)
10. 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. pp. 177–190. MLCW'05, Springer-Verlag, Berlin, Heidelberg (2006)
11. Gaona, M.A.R., Gelbukh, A.F., Bandyopadhyay, S.: Recognizing textual entailment using a machine learning approach. In: MICAI (2010)
12. Giampiccolo, D., Dang, H.T., Magnini, B., Dagan, I., Cabrio, E., Dolan, B.: The fourth PASCAL recognizing textual entailment challenge. In: Proceedings of the Fourth Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008 (2008)
13. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B.: The third pascal recognizing textual entailment challenge. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. pp. 1–9. RTE '07, Association for Computational Linguistics, Stroudsburg, PA, USA (2007)
14. Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., Shi, Y.: Recognizing textual entailment with lccâĂŹs groundhog system. In: Proceedings of the Second PASCAL Challenges Workshop. vol. 18 (2006)
15. Inkpen, D., Kipp, D., Nastase, V.: Machine learning experiments for textual entailment. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment. pp. 10–15 (2006)
16. Kozareva, Z., Montoyo, A.: Mlent: The machine learning entailment system of the university of alicante. In: Proceedings of the Second PASCAL Challenges Workshop on Recognizing Textual Entailment. pp. 10–15 (2006)
17. Landis, J., Koch, G.: The measurement of observer agreement for categorical data. Biometrics pp. 159–174 (1977)
18. Lloret, E., Ferrández, Ó., Muñoz, R., Palomar, M.: A text summarization approach under the influence of textual entailment. In: NLPCS (2008)
19. Malakasiotis, P., Androutsopoulos, I.: Learning textual entailment using svms and string similarity measures. In: Proceedings of the ACL-PASCAL Workshop

on Textual Entailment and Paraphrasing. pp. 42–47. RTE '07, Association for Computational Linguistics, Stroudsburg, PA, USA (2007)

20. Marelli, M., Menini, S., Baroni, M., Bentivogli, L., bernardi, R., Zamparelli, R.: A sick cure for the evaluation of compositional distributional semantic models. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014). European Language Resources Association (ELRA) (2014)
21. Marzelou, E., Zourari, M., Giouli, V., Piperidis, S.: Building a greek corpus for textual entailment. In: LREC (2008)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations (ICLR 2013) (2013), http://arxiv.org/abs/1301.3781
23. Negri, M., Kouylekov, M., Magnini, B.: Detecting expected answer relations through textual entailment. In: Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing. pp. 532–543. CICLing'08, Springer-Verlag, Berlin, Heidelberg (2008)
24. Padó, S., Galley, M., Jurafsky, D., Manning, C.: 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: Volume 1-Volume 1. pp. 297–305. Association for Computational Linguistics (2009)
25. Peñas, A., Rodrigo, A., Verdejo, F.: Sparte, a test suite for recognising textual entailment in spanish. In: Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Text Processing. pp. 275–286. CICLing'06, Springer-Verlag, Berlin, Heidelberg (2006)
26. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kociský, T., Blunsom, P.: Reasoning about entailment with neural attention. In: Proceedings of the International Conference on Learning Representations (ICLR 2016) (2016), http://arxiv.org/abs/1509.06664
27. 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. pp. 371–378. HLT '05, Association for Computational Linguistics, Stroudsburg, PA, USA (2005)
28. 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. pp. 1442–1451. Association for Computational Linguistics, San Diego, California (June 2016)
29. Zanzotto, F.M., Pennacchiotti, M.: Expanding textual entailment corpora fromwikipedia using co-training. In: Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources. pp. 28–36. Coling 2010 Organizing Committee, Beijing, China (August 2010)
30. Zanzotto, F.M., Pennacchiotti, M., Moschitti, A.: A machine learning approach to textual entailment recognition. Natural Language Engineering 15(4), 551–582 (Oct 2009)
31. Zeller, B., Padó, S.: A Search Task Dataset for German Textual Entailment . In: Proceedings of the 10th International Conference on Computational Semantics (IWCS). pp. 288–299. Potsdam (2013)