Entailment-based Fully Automatic Technique for Evaluation of Summaries

Pinaki Bhaskar,[1] Partha Pakray,[2] Alexander Gelbukh,[3] Sivaji Bandyopadhyay[1]

[1] Department of Computer Science and Engineering,
Jadavpur University, Kolkata – 700032, India
[2] Department of Computer and Information System,
Norwegian University of Science and Technology,
Sem Sælandsvei 7-9, NO-7491, Trondheim, Norway
[3] Center for Computing Research,
National Polytechnic Institute, Mexico City, Mexico

{pinaki.bhaskar, parthapakray}@gmail.com,
gelbukh@gelbukh.com, sivaji_cse_ju@yahoo.com

Abstract. We propose a fully automatic technique for evaluating text summaries without the need to prepare the gold standard summaries manually. A standard and popular summary evaluation techniques or tools are not fully automatic; they all need some manual process or manual reference summary. Using recognizing textual entailment (TE), automatically generated summaries can be evaluated completely automatically without any manual preparation process. We use a TE system based on a combination of lexical entailment module, lexical distance module, Chunk module, Named Entity module and syntactic text entailment (TE) module. The documents are used as text (T) and summary of these documents are taken as hypothesis (H). Therefore, the more information of the document is entailed by its summary the better the summary. Comparing with the ROUGE 1.5.5 evaluation scores over TAC 2008 (formerly DUC, conducted by NIST) dataset, the proposed evaluation technique predicts the ROUGE scores with a accuracy of 98.25% with respect to ROUGE-2 and 95.65% with respect to ROUGE-SU4.

Keywords: Automatic text summarization, summary evaluation, recognizing textual entailment.

1 Introduction

Automatic summaries are usually evaluated using human generated reference summaries or some manual efforts. Summaries generated automatically from the documents are difficult to evaluate using completely automatic evaluation process or tool.

The most popular and standard summary evaluation tools are ROUGE and Pyramid. ROUGE evaluate the automated summary by comparing it with the set of human generated reference summary. Whereas Pyramid method needs to identify the nuggets manually. Both the process are very hectic and time consuming. Therefore, automatic evaluation of summary is very important when a large number of summaries are to be evaluated, especially for multi-document summaries. For summary evaluation, we have developed an automated evaluation technique based on textual entailment.

Recognizing Textual Entailment (RTE) is one of the recent research areas of Natural Language Processing (NLP). Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by the entailing "Text" (T) and the entailed "Hypothesis" (H). T entails H if the meaning of H can be inferred from the meaning of T. Textual Entailment has many applications in NLP tasks, such as Summarization, Information Extraction, Question Answering, Information Retrieval.

There have been seven Recognizing Textual Entailment (RTE) competitions from 2005 to 2011: RTE-1 (Dagan et al., 2005), RTE-2 (Bar-Haim et al., 2006), RTE-3 (Giampiccolo et al., 2007), RTE-4 (Giampiccolo et al., 2008), RTE-5 (Bentivogli et al., 2009), RTE-6 (Bentivogli et al., 2010), and RTE-7 (Bentivogli et al., 2011).

2 Related Work

Most of the approaches in textual entailment domain take Bag-of-words representation as one option, at least as a baseline system. The system by Herrera et al. (2005) obtains lexical entailment relations from WordNet[1]. The lexical unit T entails the lexical unit H if they are synonyms, Hyponyms, Multiwords, Negations and Antonyms according to WordNet or if there is a relation of similarity between them. The system accuracy was 55.8% on RTE-1 test dataset.

Kouylekov and Magnini (2005) used a tree-edit distance algorithm applied to the dependency trees of the text and the hypothesis. If the distance (i.e., the cost of the editing operations) among the two trees is below a certain threshold, empirically estimated on the training data, then an 'YES' entailment relation is assigned between the two texts. The system accuracy was 55.9% on the RTE-1 test dataset.

Based on the idea that meaning is determined by context Clarke (2006) proposed a formal definition of entailment between two sentences in the form of a conditional probability on a measure space. The system submitted in RTE-4 provided three practical implementations of this formalism: a bag of words comparison as a baseline and two methods based on analyzing sub-sequences of the sentences possibly with intervening symbols. The system accuracy was 53% on RTE-2 test dataset.

Adams et al. (2007) used linguistic features as training data for a decision tree classifier. These features were derived from the text–hypothesis pairs under examination. The system mainly used ROUGE (Recall–Oriented Understudy for Gisting Evaluation), N-gram overlap metrics, Cosine Similarity metric and WordNet based measure as features. The system accuracy was 52% on RTE-2 test dataset.

[1] http://wordnet.princeton.edu/

In RTE-3, Newman et al. (2006) presented two systems for textual entailment, both employing decision tree as a supervised learning algorithm. The first one is based primarily on the concept of lexical overlap, considering a bag of words similarity overlap measure to form a mapping of terms in the hypothesis to the source text. The accuracy of the system improved to 67% on the RTE-3 test set.

Montalvo-Huhnet al. (2008) guessed at entailment based on word similarity between the hypotheses and the text. Three kinds of comparisons were attempted: original words (with normalized dates and numbers), synonyms and antonyms. Each of the three comparisons contributes a different weight to the entailment decision. The two-way accuracy of the system was 52.6% on RTE-4 test dataset.

Litkowski's (2009) system consists solely of routines to examine the overlap of discourse entities between the texts and hypotheses. The two-way accuracy of the system was 53% on RTE-5 Main task test dataset.

Majumdarand Bhattacharyya (2010) describes a simple lexical based system that detects entailment based on word overlap between the Text and Hypothesis. The system is mainly designed to incorporate various kinds of co-referencing that occur within a document and take an active part in the event of Text Entailment. The accuracy of the system was 47.56% on RTE-6 Main Task test dataset.

Dependency tree structures of input sentences are widely used by many research groups, since it provides more information with quite good robustness and runtime than shallow parsing techniques. Basically, a dependency parsing tree contains nodes (i.e., tokens/words) and dependency relations between nodes. Some approaches simply treat it as a graph and calculate the similarity between the text and the hypothesis graphs solely based on their nodes, while some others put more emphasis on the dependency relations themselves.

The system described by Herrera et al. (2005) is based on the use of a broad-coverage parser to extract dependency relations and a module that obtains lexical entailment relations from WordNet. The work compares whether the matching of dependency tree substructures give better evidence of entailment than the matching of plain text alone. The system accuracy was 56.6% on RTE-1 test set.

The MENT (Microsoft ENTailment) (Vanderwende et al., 2006) system predicts entailment using syntactic features and a general-purpose thesaurus, in addition to an overall alignment score. MENT is based on the premise that it is easier for a syntactic system to predict false entailments. It achieved accuracy of 60.25% on RTE-2 test set.

Wangand Neumannm (2007) present a novel approach to RTE that exploits a structure-oriented sentence representation followed by a similarity function. The structural features are automatically acquired from tree skeletons that are extracted and generalized from dependency trees. The method makes use of a limited size of training data without any external knowledge bases (e.g., WordNet) or handcrafted inference rules. They achieved an accuracy of 66.9% on the RTE-3 test data.

The major idea of Varmaet al. (2009) is to find linguistic structures, termed templates that share the same anchors. Anchors are lexical elements describing the context of a sentence. Templates that are extracted from different sentences (text and hypothesis) and connect the same anchors in these sentences are assumed to entail each other. The system accuracy was 46.8% on RTE-5 test set.

Tsuchida and Ishikawa (2011) combine the entailment score calculated by lexical-level matching with the machine-learning based filtering mechanism using various features obtained from lexical-level, chunk-level and predicate argument structure-level information. In the filtering mechanism, the false positive T-H pairs that have high entailment score but do not represent entailment are discarded. The system accuracy was 48% on RTE-7 test set.

Lin and Hovy (2003) developed an automatic summary evaluation system using *n*-gram co-occurrence statistics. Following the recent adoption by the machine translation community of automatic evaluation using the BLEU/NIST scoring process, they conduct an in-depth study of a similar idea for evaluation of summaries. They showed that automatic evaluation using unigram co-occurrences between summary pairs correlates surprisingly well with human evaluations, based on various statistical metrics, while direct application of the BLEU evaluation procedure does not always give good results.

Harnly et al. (2005) also proposed an automatic summary evaluation technique by the Pyramid method. They presented an experimental system for testing automated evaluation of summaries, pre-annotated for shared information. They reduced the problem to a combination of similarity measure computation and clustering. They achieved best results with a unigram overlap similarity measure and single link clustering, which yields high correlation to manual pyramid scores ($r = 0.942$, $p = 0.01$), and shows better correlation than the *n*-gram overlap automatic approaches of the ROUGE system.

3 Textual Entailment System

In this section we describe a two-way hybrid textual entailment (TE) recognition system that uses lexical and syntactic features. The system architecture is shown in Figure 1.

The hybrid TE system used the Support Vector Machine Learning technique that uses thirty-four features for training. Five features from Lexical TE, seventeen features from Lexical distance measure and eleven features from the rule based syntactic two-way TE system were selected.

3.1 Lexical Similarity

In this subsection, the various lexical features for textual entailment are described in detail.

WordNet based Unigram Match. In this method, the various unigrams in the hypothesis for each text-hypothesis pair are checked for their presence in text. WordNet synset are identified for each of the unmatched unigrams in the hypothesis. If any synset for the hypothesis unigram matches with any synset of a word in the text then the hypothesis unigram is considered as a WordNet based unigram match.

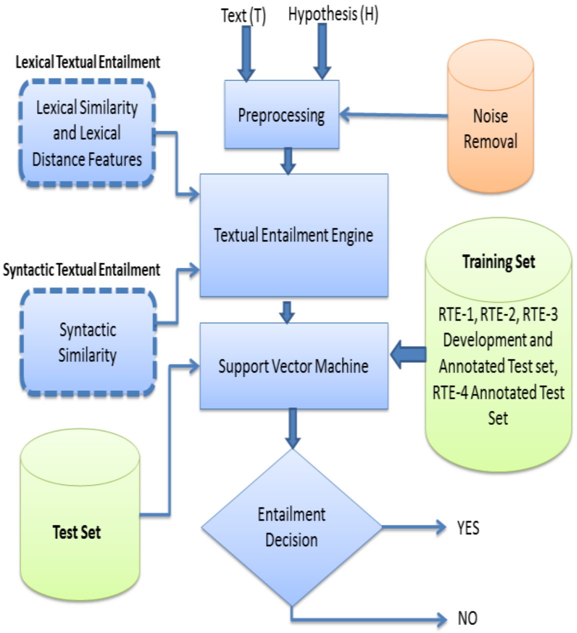
Text (T)
Hypothesis (H)
Lexical Textual Entailment
Lexical Similarity and Lexical Distance Features
Preprocessing
Noise Removal
Textual Entailment Engine
Syntactic Textual Entailment
Syntactic Similarity
Training Set
RTE-1, RTE-2, RTE-3 Development and Annotated Test set, RTE-4 Annotated Test Set
Support Vector Machine
Test Set
Entailment Decision
YES
NO

Fig. 1. Hybrid Textual Entailment System

Bigram Match. Each bigram in the hypothesis is searched for a match in the corresponding text part. The measure Bigram_Match is calculated as the fraction of the hypothesis bigrams that match in the corresponding text, i.e.,

$$\text{Bigram\_Match} = \frac{\text{Total number of matched bigrams in a text} - \text{hypothesis pair}}{\text{Number of hypothesis bigrams}}.$$

Longest Common Subsequence (LCS). The Longest Common Subsequence of a text-hypothesis pair is the longest sequence of words which is common to both the text and the hypothesis. LCS(T,H) estimates the similarity between text T and hypothesis H, as

$$\text{LCS\_Match} = \frac{\text{LCS(T, H)}}{\text{length of H}}.$$

Skip-grams. A skip-gram is any combination of n words in the order as they appear in a sentence, allowing arbitrary gaps. In the present work, only 1-skip-bigrams are considered where 1-skip-bigrams are bigrams with one word gap between two words in order in a sentence. The measure 1-skip_bigram_Match is defined as

$$\text{1\_skip\_bigram\_Match} = \frac{\text{skip\_gram(T, H)}}{n},$$

where skip_gram(T,H) refers to the number of common 1-skip-bigrams (pair of words in sentence order with one word gap) found in T and H and n is the number of 1-skip-bigrams in the hypothesis H.

Stemming. Stemming is the process of reducing terms to their root forms. For example, the plural forms of a noun such as 'boxes' are stemmed into 'box', and inflectional endings with 'ing', 'es', 's' and 'ed' are removed from verbs. Each word in the text and hypothesis pair is stemmed using the stemming function provided along with the WordNet 2.0.

If s_1 is the number of common stemmed unigrams between text and hypothesis and s_2 is the number of stemmed unigrams in Hypothesis, then the measure Stemming_Match is defined as

$$\text{Stemming\_Match} = \frac{s_1}{s_2}.$$

WordNet is one of most important resource for lexical analysis. WordNet 2.0 has been used for WordNet based unigram match and stemming step. API for WordNet Searching[2] (JAWS) is an API that provides Java applications with the ability to retrieve data from the WordNet database.

3.2 Syntactic Similarity

In this section, various syntactic similarity features for textual entailment are described in detail. This module is based on the Stanford Dependency Parser[3], which normalizes data from the corpus of text and hypothesis pairs, accomplishes the dependency analysis and creates appropriate structures Our Entailment system uses the following features.

Subject. The dependency parser generates nsubj (nominal subject) and nsubjpass (passive nominal subject) tags for the subject feature. Our entailment system uses these tags.

Object. The dependency parser generates dobj (direct object) as object tags.

Verb. Verbs are wrapped with either the subject or the object.

Noun. The dependency parser generates NN (noun compound modifier) as noun tags.

Preposition. Different types of prepositional tags are prep_in, prep_to, prep_with etc. For example, in the sentence "A plane crashes in Italy." the prepositional tag is identified as prep_in(in, Italy).

[2] http://wordnetweb.princeton.edu/perl/webwn

[3] http://www-nlp.stanford.edu/software/lex-parser.shtml

Determiner. Determiner denotes a relation with a noun phase. The dependency parser generates det as determiner tags. For example, the parsing of the sentence “A journalist reports on his own murders.” generates the determiner relation as det(journalist,A).

Number. The numeric modifier of a noun phrase is any number phrase. The dependency parser generates num (numeric modifier). For example, the parsing of the sentence “Nigeria seizes 80 tonnes of drugs.” generates the relation num (tonnes, 80).

Matching Module. After dependency relations are identified for both the text and the hypothesis in each pair, the hypothesis relations are compared with the text relations. The different features that are compared are noted below. In all the comparisons, a matching score of 1 is considered when the complete dependency relation along with all of its arguments matches in both the text and the hypothesis. In case of a partial match for a dependency relation, a matching score of 0.5 is assumed.

Subject-Verb Comparison. The system compares hypothesis subject and verb with text subject and verb that are identified through the nsubj and nsubjpass dependency relations. A matching score of 1 is assigned in case of a complete match. Otherwise, the system considers the following matching process.

WordNet Based Subject-Verb Comparison. If the corresponding hypothesis and text subjects do match in the subject-verb comparison, but the verbs do not match, then the WordNet distance between the hypothesis and the text is compared. If the value of the WordNet distance is less than 0.5, indicating a closeness of the corresponding verbs, then a match is considered and a matching score of 0.5 is assigned. Otherwise, the subject-subject comparison process is applied.

Subject-Subject Comparison. The system compares hypothesis subject with text subject. If a match is found, a score of 0.5 is assigned to the match.

Object-Verb Comparison. The system compares hypothesis object and verb with text object and verb that are identified through DObj dependency relation. In case of a match, a matching score of 0.5 is assigned.

WordNet Based Object-Verb Comparison. The system compares hypothesis object with text object. If a match is found then the verb associated with the hypothesis object is compared with the verb associated with the with text object. If the two verbs do not match then the WordNet distance between the two verbs is calculated. If the value of WordNet distance is below 0.50 then a matching score of 0.5 is assigned.

Cross Subject-Object Comparison. The system compares hypothesis subject and verb with text object and verb or hypothesis object and verb with text subject and verb. In case of a match, a matching score of 0.5 is assigned.

Number Comparison. The system compares numbers along with units in the hypothesis with similar numbers along with units in the text. Units are first compared and if they match then the corresponding numbers are compared. In case of a match, a matching score of 1 is assigned.

Noun Comparison. The system compares hypothesis noun words with text noun words that are identified through NN dependency relation. In case of a match, a matching score of 1 is assigned.

Prepositional Phrase Comparison. The system compares the prepositional dependency relations in the hypothesis with the corresponding relations in the text and then checks for the noun words that are arguments of the relation. In case of a match, a matching score of 1 is assigned.

Determiner Comparison. The system compares the determiners in the hypothesis and in the text that are identified through Det relation. In case of a match, a matching score of 1 is assigned.

Other relation Comparison. Besides the above relations that are compared, all other remaining relations are compared verbatim in the hypothesis and in the text. In case of a match, a matching score of 1 is assigned.

3.3 Part-of-Speech (POS) Matching

This module basically matches common POS tags between the text and the hypothesis pairs. Stanford POS tagger[4] is used to tag the part of speech in both text and hypothesis. System matches the verb and noun POS words in the hypothesis with those in the text. A score POS_match is defined as follows:

$$\text{POS\_Match} = \frac{\text{number of verb and noun matched in Text and Hypothesis}}{\text{total number of verbs and nouns in Hypothesis}}. \quad (1)$$

3.4 Lexical Distance

The important lexical distance measures that are used in the present system include Vector Space Measures (Euclidean distance, Manhattan distance, Minkowsky distance, Cosine similarity, Matching coefficient), Set-based Similarities (Dice, Jaccard, Overlap, Cosine, Harmonic), Soft-Cardinality, Q-Grams Distance, Edit Distance Measures (Levenshtein distance, Smith-Waterman Distance, Jaro).

[4] http://nlp.stanford.edu/software/tagger.shtml

3.5 Chunk similarity

The part of speech (POS) tags of the hypothesis and text are identified using the Stanford POS tagger. After getting the POS information, the system extracts the chunk output using the CRFChunker[5]. Chunk boundary detector detects each individual chunk such as noun chunk, verb chunk etc. Thus, all the chunks for each sentence in the hypothesis are identified. Each chunk of the hypothesis is now searched in the text side and the sentences that contain the key chunk words are extracted. If chunks match then the system assigns scores for each individual chunk corresponding to the hypothesis. The scoring values are changed according to the matching of chunk and word containing the chunk. The entire scoring calculation is given in (2) and (3):

$$\text{Match score } M[i] = \frac{W_m[i]}{W_c[i]}, \tag{2}$$

where $W_m\,[i]$ is the number of words that match in the i-th chunk and $W_c\,[i]$ is the total number of words containing the i-th chunk;

$$\text{Overall score S} = \sum_{i=1}^{N} \frac{M\ [i]}{\mathrm{N}}. \tag{3}$$

where N is the total number of chunks in the hypothesis.

3.6 Support Vector Machines (SVM).

In machine learning, support vector machines (SVMs)[6] are supervised learning models used for classification and regression analysis. Associated learning algorithms analyze data and recognize patterns. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes form the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

The LIBSVM[7] tool was used to find the textual entailment relation. The system has used LIBSVM for building the model file. The TE system has used the following data sets: RTE-1 development and test set, RTE-2 development and annotated test set, RTE-3 development and annotated test set and RTE-4 annotated test set to deal with the two-way classification task for training purpose to build the model file. The LIBSVM tool is used by the SVM classifier to learn from this data set. For training purpose, 3967

[5] http://crfchunker.sourceforge.net/

[6] http://en.wikipedia.org/wiki/Support_vector_machine

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

text-hypothesis pairs were used. It has been tested on the RTE test dataset and we obtained 60% to 70% accuracy on RTE datasets. We have applied this textual entailment system on summarize data sets and system gives the entailment score with entailment decisions (i.e., "YES" / "NO"). We have tested in both directions.

4 Automatic Evaluation of Summary

Ideally, summary of some documents should contain all the necessary information contained in the documents. So the quality of a summary should be judged on how much information of the documents it contains. If the summary contains all the necessary information from the documents, then it will be a perfect summary. Yet manual comparison is the best way to judge that how much information it contains from the document. However, manual evaluation is a very hectic process, especially when the summary generated from multiple documents. When a large number of multi-document summaries have to be evaluated, then an automatic evaluation method needs to evaluate the summaries. Here we propose textual entailment (TE) based automatic evaluation technique for summary.

4.1 Textual Entailment based Summary Evaluation

Textual Entailment is defined as a directional relationship between pairs of text expressions, denoted by the entailing "Text" (T) and the entailed "Hypothesis" (H). Text (T) entails hypothesis (H) if the information of text (T) is inferred into the hypothesis (H). Here the documents are used as text (T) and summary of these documents are taken as hypothesis (H). Therefore, if the information of documents is entailed into the summary then it will be a very good summary, which should get a good evaluation score.

As our textual entailment system works on sentence level each sentence of documents are taken as text (T) and calculate the entailment score comparing with each sentence of the summary assuming them as hypothesis (H). For example, if T_i is the i^{th} sentence of documents, then it will compared with each sentence of the summary, i.e. H_j, where $j = 1$ to n; and n is the total number of sentences in the summary. Now if T_i is validated with any one of the summary sentences using our textual entailment system, then it will be marked as validated. After get the entailment result of all the sentences of documents, the percentage or ratio of the marked/validated sentences with respect to unmarked / rejected sentences will be the evaluation score of the summary.

5 Data Collection

We used the Text Analysis Conference (TAC, formerly DUC, conducted by NIST) 2008 Update Summarization track's datasets8 for this experiment. This dataset contains 48 topics and each topic has two sets of 10 documents, i.e. there are 960 documents.

[8] http://www.nist.gov/tac/data/index.html

The evaluation data set has four model summaries for each document set, i.e. 8 model summaries for each topic. In 2008, there are 72 participants, and we used the summaries of all the participants of this year.

6 Comparison of Automatic vs. Manual Evaluation

We considered the evaluation scores of all the 72 participants of TAC 2008 using ROUGE 1.5.5. We calculated the evaluation scores of the same summaries of 72 participants using the proposed automated evaluation technique and compared it with ROUGE scores. The comparison of the evaluation scores on top five participants is shown in the Table 1.

Table 1. Comparison of Summary Evaluation Score

Evaluation method	ROUGE-2 Average_R	ROUGE-SU4 Average_R	Proposed method
Top ranked participant (id:43)	0.111	0.143	0.7063
2nd ranked participant (id:13)	0.110	0.140	0.7015
3rd ranked participant (id:60)	0.104	0.142	0.6750
4th ranked participant (id:37)	0.103	0.143	0.6810
5th ranked participant (id:6)	0.101	0.140	0.6325

For measuring the accuracy of our proposed method, we consider the ROUGE 1.5.5 evaluation score as the gold standard score and then calculate the accuracy of this proposed method using (4):

$$\text{Accuracy} = 1 - \frac{\sum_{i=1}^{n} |(r_i - r_i^R)|}{n^2} \qquad (4)$$

where r_i is the rank of i-th summary after evaluated by the proposed method, r_i^R is the rank of i-th summary after evaluated by ROUGE 1.5.5, and n is the total number of multi-document summaries.

After evaluating 48 (only set A) multi-document summaries of 72 participants, i.e total 3456 multi-document summaries using the evaluation method, ROUGE 1.5.5 and the proposed method, the accuracy of this proposed method calculated using (4) comparing with the ROUGE's evaluation scores. The accuracy figures are 0.9825 with respect to ROUGE-2 and 0.9565 with respect to ROUGE-SU4.

7 Conclusions

Evaluating summaries automatically is very useful in batch processing. From the comparison of evaluation scores of the proposed method and those of ROUGE 1.5.5, it is clear that our method can predict the ROUGE ranking. However, ROUGE requires manually preparing the gold standard summaries, which is a very time-consuming task. In contrast, our method is completely automatic.

In our future work, we plan to explore the use of syntactic *n*-grams, which have been shown to be useful on other automatic evaluation tasks (Sidorov et al., 2013).

Acknowledgments. We acknowledge the support of the DST India-CONACYT Mexico project "Answer Validation through Textual Entailment" funded by DST, Government of India and the Department of Electronics and Information Technology (DeitY), Ministry of Communications & Information Technology (MCIT), Government of India funded project "Development of Cross Lingual Information Access (CLIA) System Phase II", and SIP-IPN project 20131702. We also acknowledge the support of the Department of Computer and Information Science, Norwegian University of Science and Technology, and ABCDE Fellowship Programme 2012-1013.

References

1. Adams, R., Nicolae, G., Nicolae, C. and Harabagiu, S. (2007): Textual Entailment Through Extended Lexical Overlap and Lexico-Semantic Matching. In Proceedings of the ACL PASCAL Workshop on Textual Entailment and Paraphrasing. 28–29 June, Prague, Czech Republic, pp. 119–124.
2. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I. (2006): The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy.
3. Bentivogli, L., Dagan, I., Dang. H.T., Giampiccolo, D., Magnini, B. (2009): The Fifth PASCAL Recognizing Textual Entailment Challenge, In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA.
4. Bentivogli, L., Clark, P., Dagan, I., Dang, H. T., Giampiccolo, D. (2010): The Sixth PASCAL Recognizing Textual Entailment Challenge. In TAC 2010 Notebook Proceedings.
5. Bentivogli, L., Clark, P., Dagan, I., Dang, H. T., Giampiccolo, D. (2009): The Seventh PASCAL Recognizing Textual Entailment Challenge. In TAC 2011 Notebook Proceedings.
6. Clarke, D. (2006): Meaning as Context and Subsequence Analysis for Entailment. In Proceedings of the Second PASCAL Recognising Textual Entailment Challenge, Venice, Italy.
7. Dagan, I., Glickman, O., Magnini, B. (2005): The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop.
8. Giampiccolo, D., Magnini, B., Dagan, I., Dolan, B. (2007): The Third PASCAL Recognizing Textual Entailment Challenge, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic.
9. Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E. (2008): The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings.
10. Harnly, A., Nenkova, A., Passonneau, R., Rambow, O. (2005): Automation of summary evaluation by the pyramid method. Recent Advances in Natural Language Processing (RANLP). Borovets, Bulgaria.
11. Herrera, J., Peas, A. Verdejo, F. (2005): Textual Entailment Recognition Based on Dependency Analysis and WordNet. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 21-24, 33–36 April 2005, Southampton, U.K.
12. Kouylekov, M., Magnini, B. (2005): Recognizing Textual Entailment with Tree Edit Distance Algorithms. Proceedings of the First PASCAL Recognizing Textual Entailment Workshop.

13. Lin, C. Y., Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Volume 1. Association for Computational Linguistics, pp. 71–78.
14. Litkowski, K. (2009): Overlap Analysis in Textual Entailment Recognition. In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA.
15. Majumdar, D. Bhattacharyya, P. (2010): Lexical Based Text Entailment System for Summarization Settings of RTE6.Proceedings of the Text Analysis Conference (TAC 2010) November 15–16, 2010 National Institute of Standards and Technology Gaithersburg, Maryland, USA.
16. Montalvo-Huhn, O. Taylor, S. (2008): Textual Entailment – Fitchburg State College. In Proceedings of TAC08, Fourth PASCAL Challenges Workshop on Recognising Textual Entailment.
17. Newman, E., Dunnion, J., Carthy, J. (2006): Constructing a Decision Tree Classifier using Lexical and Syntactic Features. In Proceedings of the Second PASCAL Recognising Textual Entailment Challenge.
18. Sidorov, G., Gupta, A., Tozer, M., Catala, D., Catena, A., Fuentes, S (2013): Rule-based System for Automatic Grammar Correction Using Syntactic N-grams for English Language Learning (L2). In: Proc. ACL 2013, 6 p.
19. Tsuchida, M., Ishikawa, K. (2011): IKOMA at TAC2011: A Method for Recognizing Textual Entailment using Lexical-level and Sentence Structure-level features. In TAC 2011 Notebook Proceedings.
20. Vanderwende, L., Menezes, A., Snow, R. (2006): Microsoft Research at RTE-2: Syntactic Contributions in the Entailment Task: an implementation. In Proceedings of the Second PASCAL Challenges Workshop.
21. Varma, V., Bharat, V., Kovelamudi, S., Bysani, P., GSK, S., N, K. K., Reddy, K., Kumar, K., Maganti, N. (2009): IIIT Hyderabad at TAC 2009. In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA.
22. Wang, R., Neumannm G. (2007): Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons. In Proceedings of the Third PASCAL Recognising Textual Entailment Challenge.