Automatic Evaluation of Summary Using Textual Entailment

Pinaki Bhaskar
Department of Computer Science &
Engineering, Jadavpur University,
Kolkata – 700032, India
pinaki.bhaskar@gmail.com

Partha Pakray
Department of Computer Science &
Engineering, Jadavpur University,
Kolkata – 700032, India
parthapakray@gmail.com

Abstract

This paper describes about an automatic technique of evaluating summary. The standard and popular summary evaluation techniques or tools are not fully automatic; they all need some manual process. Using textual entailment (TE) the generated summary can be evaluated automatically without any manual evaluation/process. The TE system is the composition of lexical entailment module, lexical distance module, Chunk module, Named Entity module and syntactic text entailment (TE) module. The syntactic TE system is based on the Support Vector Machine (SVM) that uses twenty five features for lexical similarity, the output tag from a rule based syntactic two-way TE system as a feature and the outputs from a rule based Chunk Module and Named Entity Module as the other features. The documents are used as text (T) and summary of these documents are taken as hypothesis (H). So, if the information of documents is entailed into the summary then it will be a very good summary. After comparing with the ROUGE 1.5.5 evaluation scores, the proposed evaluation technique achieved a high accuracy of 98.25% w.r.t ROUGE-2 and 95.65% w.r.t ROUGE-SU4.

1 Introduction

Automatic summaries are usually evaluated using human generated reference summaries or some manual efforts. The summary, which has been generated automatically from the documents, is difficult to evaluated using completely automatic evaluation process or tool. The most popular and standard summary evaluation tool is ROUGE and Pyramid. ROUGE evaluates the automated summary by comparing it with the set of human generated reference summary. Where as Pyramid method needs to identify the nuggets manually. Both the processes are very hectic and time consuming. So, automatic evaluation of summary is very much needed when a large number of summaries have to be evaluated, specially 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.

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 (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.

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 subsequences of the sentences possibly with intervening symbols. The system accuracy was 53% on RTE-2 test dataset.

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

Proceedings of the Student Research Workshop associated with RANLP 2013, pages 30–37,
Hissar, Bulgaria, 9-11 September 2013.

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

Montalvo-Huhn et 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.

Majumdar and Bhattacharyya (2010) describe a simple lexical based system, which 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.

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. The system accuracy was 60.25% on RTE-2 test set.

Wang and 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 Varma et 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 evaluating summaries. They showed that automatic evaluation using unigram co-occurrences between summary pairs correlates surprising 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

A two-way hybrid textual entailment (TE) recognition system that uses lexical and syntactic features has been described in this section. The system architecture has been shown in Figure 1. The hybrid TE system as (Pakray et al., 2011b) has 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 have been selected.

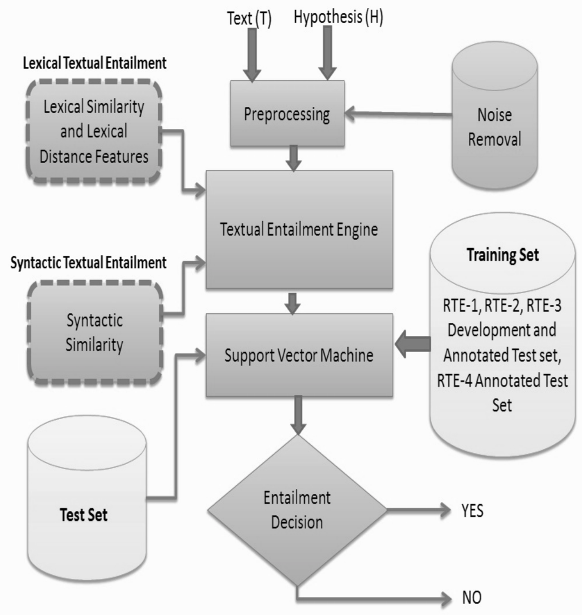
Text (T)
Hypothesis (H)
Lexical Textual Entailment
Lexical Similarity and Lexical Distance Features
Preprocessing
Noise Removal
Textual Entailment Engine
Training Set
RTE-1, RTE-2, RTE-3 Development and Annotated Test set, RTE-4 Annotated Test Set
Syntactic Textual Entailment
Syntactic Similarity
Support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

Figure 1. Hybrid Textual Entailment System

3.1 Lexical Similarity

In this section the various lexical features (Pakray et al., 2011b) for textual entailment are described in detail.

i. 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.

ii. 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., Bigram_Match = (Total number of matched bigrams in a text-hypothesis pair /Number of hypothesis bigrams).

iii. 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 LCS_Match=LCS(T,H)/length of H.

iv. 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

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

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.

v. 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 s1= number of common stemmed unigrams between text and hypothesis and s2= number of stemmed unigrams in Hypothesis, then the measure Stemming_match is defined as Stemming_Match=s1/s2

WordNet is one of most important resource for lexical analysis. The 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 the various syntactic similarity features (Pakray et al., 2011b) 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.

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

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

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

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

e. Preposition. Different types of prepositional tags are prep_in, prep_to, prep_with etc. For example, in the sentence “A plane crashes in Ita-

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

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

ly.” the prepositional tag is identified as prep_in(in, Italy).

f. 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).

g. 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.

i. Subject-Verb Comparison. The system compares hypothesis subject and verb with text subject and verb that are identified thROUGE 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.

ii. 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.

iii. 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.

iv. 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.

v. 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.

vi. 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.

vii. 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.

viii.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.

ix. 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.

x. 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.

xi. 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 in equation 2.

$$\text{POS\_Match} = \frac{\text{Number of Verb and Noun Match in Text and Hypothesis}}{\text{Total number of Verb and Noun in Hypothesis}} \qquad (2)$$

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

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). These lexical distance features have been used as described in detail by Pakray et al. (2011b).

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 CRF Chunker[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 equations 3 and 4 below:

$$\text{Match score } (M[i]) = \frac{W_m[i]}{W_c[i]} \qquad (3)$$

where, $W_m[i]$ = Number of words that match in the i^{th} chunk and $W_c[i]$ = Total number of words containing the i^{th} chunk.

$$\text{Overall score } (S) = \sum_{i=1}^{N} \frac{M[i]}{N} \qquad (4)$$

where, N = 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 system has used LIBSVM[7] 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 text-hypothesis pairs have been used. It has been tested on the RTE test dataset and we have got 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. But manual comparison is the best way to judge that how much information it contains from the document. But manual evaluation is a very hectic process, specially 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 (TE) Based Summary Evaluation

Textual Entailment is defined as a directional relationship between pairs of text expressions,

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

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

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