TEXTUAL ENTAILMENT USING LEXICAL AND SYNTACTIC SIMILARITY

Partha Pakray[1], Sivaji Bandyopadhyay[1] and Alexander Gelbukh[2]

[1]Department of Computer Science & Engineering,
Jadavpur University, Kolkata, India.
parthapakray@gmail.com
sbandyopadhyay@cse.jdvu.ac.in
[2]Center for Computing Research, National Polytechnic Institute,
gelbukh@gelbukh.com

ABSTRACT

A two-way Textual Entailment (TE) recognition system that uses lexical and syntactic features has been described in this paper. The TE system is rule based that uses lexical and syntactic similarities. The important lexical similarity features that are used in the present system are: WordNet based uni-gram match, bi-gram match, longest common sub-sequence, skip-gram, stemming. In the syntactic TE system, the important features used are: subject-subject comparison, subject-verb comparison, object-verb comparison and cross subject-verb comparison. The system has been separately trained on each development corpus released as part of the Recognising Textual Entailment (RTE) competitions RTE-1, RTE-2, RTE-3 and RTE-5 and tested on the respective RTE test sets. No separate development data was released in RTE-4. The evaluation results on each test set are compared with the RTE systems that participated in the respective RTE competitions with lexical and syntactic approaches.

KEYWORDS

Textual Entailment (TE), Lexical Similarity, Syntactic Similarity, Dependency Parsing, RTE data sets, System Evaluation.

1. INTRODUCTION

Recognizing Textual Entailment (RTE) is one of the recent challenges 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 Natural Language Processing (NLP) tasks, such as : in Summarization (SUM), a summary should be entailed by the text; Paraphrases (PP) can be seen as mutual entailment between a text T and a hypothesis H; in Information Extraction (IE), the extracted information should also be entailed by the text; in Question Answering (QA) the answer obtained for one question after the Information Retrieval (IR) process must be entailed by the supporting snippet of text.

There were three Recognizing Textual Entailment competitions RTE-1 in 2005 [1], RTE-2 [2] in 2006 and RTE-3 [3] in 2007 which were organized by PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning) - the European Commission's IST-funded Network of Excellence for Multimodal Interfaces. In 2008, the fourth edition (RTE-4) [4] of the challenge was organized by NIST (National Institute of Standards and Technology) in Text Analysis Conference (TAC). The TAC RTE-5 [5] challenge in 2009 includes a separate search pilot along with the main task. The TAC RTE-6 challenge [6], in 2010, includes the Main Task and

DOI : 10.5121/ijaia.2011.2104

Novelty Detection Task along with RTE-6 KBP Validation Pilot Task. The RTE-6 does not include the traditional RTE Main Task which was carried out in the first five RTE challenges, i.e., no tasks are defined to make entailment judgements over isolated T-H pairs drawn from multiple applications. In every new competition several new features of RTE were introduced. In 2010, Parser Training and Evaluation using Textual Entailment [7] was organized by SemEval-2. We have developed our own RTE system and have participated in RTE-5 in 2009, in the Parser Training and Evaluation using Textual Entailment as part of SemEval-2 and also in the TAC RTE-6 challenge in 2010.

The first PASCAL Recognizing Textual Entailment Challenge (RTE-1) [1] introduced the first benchmark for the entailment recognition task. The RTE-1 dataset consists of manually collected text fragment pairs, termed text (t) (1-2 sentences) and hypothesis (h) (one sentence). The systems were required to judge for each pair whether t entails h. The pairs represented success and failure settings of inferences in various application types (termed "tasks"). In RTE-1 the various techniques used by the participating systems are word overlap, WordNet, statistical lexical relation, world knowledge, syntactic matching and logical inference.

After the success of RTE-1, the main goal of the RTE-2, held in 2006 [2], was to support the continuity of research on textual entailment. The RTE-2 data set was created with the main focus of providing more "realistic" text-hypothesis pair. As in the RTE-1, the main task was to judge whether a hypothesis H is entailed by a text T. The texts in the datasets were of 1-2 sentences, while the hypotheses were one sentence long. The following four applications – Information Extraction (IE), Information Retrieval (IR), Question Answering (QA) and Summarization (SUM) were considered as settings or contexts for the generation of each pair. Again, the examples were drawn to represent different levels of entailment reasoning, such as lexical, syntactic, morphological and logical. The main task in the RTE-2 challenge was classification – entailment judgement for each pair in the test set that represented either entailment or no entailment. Accuracy, i.e., the percentage of pairs correctly judged, was defined as the evaluation criteria for the task. A secondary task was created to rank the pairs based on their entailment confidence. All positive pairs (for which the entailment holds) are placed before all negative pairs in a perfect ranking. This task was evaluated using the average precision measure [8], which is a common evaluation measure used for ranking in information retrieval. In RTE-2 the techniques used by the various participating systems are Lexical Relation/ database, n-gram / subsequence overlap, syntactic matching / Alignment, Semantic Role labelling/ FrameNet / PropBank, Logical Inference, Corpus / web-based statistics, machine learning (ML) Classification, Paraphrase and Templates, Background Knowledge and acquisition of entailment corpus.

The RTE-3 data set consisted of 1600 text-hypothesis pairs, equally divided into a development set and a test set. The four applications from RTE-2, i.e., IE, IR, QA and SUM, were set as the contexts for the generation of each pair. 200 pairs were selected for each application in each data set. Each pair was annotated with its related task (IE/IR/QA/SUM) and entailment judgement (YES/NO). In addition, an optional pilot task, called "Extending the Evaluation of Inferences from Texts" was set up by the NIST, in order to explore two other sub-tasks closely related to textual entailment: differentiating unknown entailment from identified contradictions and providing justifications for system decisions. The idea in the first sub-task was to drive systems that make more precise informational distinctions, taking a three-way decision between "YES", "NO" and "UNKNOWN". Thus, a hypothesis being unknown on the basis of a text would be distinguished from a hypothesis being shown false/contradicted by a text.

In RTE-4, [4], no development set was provided, as the pairs proposed were very similar to the ones contained in RTE-3 development and test sets, which could therefore be used to train the

systems. Four applications, i.e., IE, IR, QA and SUM, were set as the contexts for the generation of each pair. The length of the Hypotheses was the same as in the past data sets (RTE-3); however, the Texts were generally longer. A major difference with respect to RTE-3 was that the RTE-4 data set consisted of 1000 T-H pairs, instead of 800.

In RTE-4, the challenges were classified as two-way task and three-way task. The two-way RTE task was to decide whether:
i. T entails H - the pair would be marked as ENTAILMENT;
ii. T does not entail H - the pair would be marked as NO ENTAILMENT.

The three-way RTE task was to decide whether:
i. T entails H - the pair would be marked as ENTAILMENT
ii. T contradicts H - the pair would be marked as CONTRADICTION
iii. The truth of H could not be determined on the basis of T - the pair would be marked as UNKNOWN.

The structure of the RTE-5 [5] Main Task remained unchanged, offering both the traditional two-way task and the three-way task introduced in the previous campaign. Moreover, a pilot Search Task was set up in order to find all the sentences in a set of documents that entail a given hypothesis. An important innovation introduced in this campaign was mandatory ablation tests that participants had to perform for all major knowledge resources employed by the respective participating systems.

A major innovation was introduced in The RTE-6 Challenge [6]. The traditional Main Task was replaced by a new task, similar to the RTE-5 Search Pilot, in which Textual Entailment is performed on a real corpus in the Update Summarization scenario. A subtask was also proposed, aimed at detecting novel information. To continue the effort of testing RTE in NLP applications, a KBP Validation Pilot Task was set up, in which RTE systems had to validate the output of systems participating in the KBP Slot Filling Task.

We participated in TAC RTE-5, TAC RTE-6 Challenge and SemEval-2 Parser Training and Evaluation using Textual Entailment Task. In the present paper, a 2-way lexical and syntactic textual entailment recognition system has been described. Related works are described in Section 2. The RTE system is described in Section 3 that includes detailed discussions on the lexical and syntactic similarity approaches. The various experiment carried out on the development and test data sets are described in Section 4 along with the results. In Section 5, the experimental results are compared with the RTE systems based on lexical and syntactic similarity approaches participating in the respective RTE competitions. The conclusions are drawn in Section 6.

2. RELATED WORKS

In the various RTE Challenges, several methods are applied on the textual entailment task. Most of these systems use some sort of lexical matching, e.g., n-gram, word similarity etc. and even simple word overlap. A number of systems represent the texts as parse trees (e.g., syntactic or dependency trees) before the actual task. Some of the systems use semantic relation (e.g., logical inference, Semantic Role Labelling) for solving the text and hypothesis entailment problem.

The system [9] investigate two new models for the RTE problem that employ simple generic Bracketing Inversion Transduction Grammar (ITG). The CLaC Lab's system [10] for the PASCAL RTE challenge explores the potential of simple general heuristics and a knowledge-

poor approach for recognising paraphrases, using NP coreference, NP chunking, and two parsers (RASP and Link) to produce Predicate Argument Structures (PAS) for each of the pair components. WordNet lexical chains and a few specialised heuristics were used to establish semantic similarity between corresponding components of the PAS from the pair. The system [11] defined a measure for textual entailment recognition based on the graph matching theory applied to syntactic graphs. They described the experiments carried out to estimate measure's parameters with SVM. The system [12] combined two methods to tackle the textual entailment challenge: a shallow method based on word overlap and a deep method using theorem proving techniques. They used a machine learning technique to combine features derived from both methods. The UNED-NLP Group Recognizing Textual Entailment System [13] was based on the use of a broad-coverage parser to extract dependency relations and a module which obtains lexical entailment relations from WordNet. The work aims at comparing whether the matching of dependency trees substructures give better evidence of entailment than the matching of plain text alone.

The system [14] reports the description of the developed system and the results obtained in the participation of the UNED in the Second Recognizing Textual Entailment (RTE) Challenge. New techniques and tools have been added: enriched queries to WordNet, detection of numeric expressions and their entailment, and Support Vector Machine classification (SVM) were the more relevant. The system [15] described a machine learning based approach for the resolution of text entailment. Their model features based on lexical overlaps and semantic similarity measures. The machine learning algorithm they worked with is Support Vector Machines. Several feature sets are constructed and their combination is studied in order to boost the final performance of the MLEnt system. The system [16] that used machine learning algorithms to combine features that capture various shallow heuristics for the task of recognizing textual entailment. The features quantify several types of matches and mismatches between the test and hypothesis sentences. Matching features represent lexical matching (including synonyms and related words), part-of-speech matching and matching of grammatical dependency relations. Mismatch features include negation and numeric mismatches. The system [17] used a word-based similarity combined with a tree-based similarity approach. The System [18] was estimating the cost of the information of the hypothesis which is missing in the text and can not be matched with entailment rules. They have tested different system settings for calculating the importance of the words of the hypothesis and investigated the possibility of combining them with machine learning algorithm. The system described in [19] consists of a bag of words similarity overlap measure, derived from a combination of WordNet lexical chains to form a mapping of terms in the hypothesis to the source text. These items were entered into a decision tree to determine the overall entailment relation.

The system [20] introduced a system for textual entailment that is based on a probabilistic model of entailment. The model is defined using some calculus of transformations on dependency trees, which is characterized by the fact that derivations in that calculus preserve the truth only with a certain probability. The system [21] used knowledge such as gazetteers, WordNet and custom built knowledge bases are also likely to improve performance, their goal is to characterize the syntactic features alone that aid in accurate entailment prediction. The system [22] used a Machine Learning approach with Support Vector Machines and AdaBoost to deal with the RTE challenge. They performed a lexical, syntactic, and semantic analysis of the entailment pairs. From this information they compute a set of semantic based distances between sentences. The system [23] based on the core approach of the tree edit distance algorithm, the system central module was designed to target the scope of TE semantic variability. The main idea was to transform the hypothesis making use of extensive semantic knowledge from sources like DIRT, WordNet, Wikipedia, acronyms database. Additionally, they built a system to acquire the extra background knowledge needed and applied complex grammar rules for

rephrasing in English. The system [24] used Lexical relations, WordNet and Syntactic Matching for solving the textual entailment problem. The system presented in [25] proposed 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 system described in [26] use four Support Vector Machines, one for each sub task of the challenge, with features that correspond to string similarity measures operating at the lexical and shallow syntactic level.

The Emory system [27] used a supervised machine learning approach to train a classifier over a variety of lexical, syntactic, and semantic metrics. The system [28] proposed an unsupervised similarity metric to measure the relevance of word pairs using the Web1T data. The alignment scores between the dependency trees of the text and the hypothesis sentences are calculated based on this new similarity metric and these scores are used to predict the entailment between the text and the hypothesis sentences. The system [29] based on word similarity between the hypotheses and the text. They attempt three kinds of comparisons: original words (with normalized dates and numbers) synonyms, and antonyms. Each of the three comparisons contributes a different weight to the entailment decision. The system [30] present a new data structure, termed compact forest, which allows efficient generation and representation of entailed consequents, each represented as a parse tree. Rule-based inference is complemented with a new approximate matching measure inspired by tree kernels, which is computed efficiently over compact forests. Their system also makes use of novel large-scale entailment rule bases, derived from Wikipedia as well as from information about predicates and their argument mapping, gathered from available lexicons and complemented by unsupervised learning. A syntactic dependency tree approach for the task of textual entailment is used in [31]. The system approach is to construct the syntactic dependency trees for both text and hypothesis sentences and then compare the nodes of the dependency trees by using the semantic similarity between the two nodes. The system described in [32] applied a Support Vector Machine classifier to examples characterized by four features that are based on: edit distance, distance in WordNet and Longest Common Sub-string between text and hypothesis.

The system [33] used string similarity measures applied to shallow abstractions of the input sentences, and a Maximum Entropy classifier to learn how to combine the resulting features. They also exploited WordNet to detect synonyms and a dependency parser to measure similarity in the grammatical structure of T and H. The system [34] is based on the composition of the following six lexical based RTE methods: WordNet based unigram match, bigram match, longest common sub-sequence, skip-gram, stemming and named entity matching. Each of these methods were applied on the development data to obtain two-way decisions. The system [35] is based on the lexical entailment between two text excerpts, namely the hypothesis and the text. To extract atomic parts of hypotheses and texts, we carry out syntactic parsing on the sentences. We then utilize WordNet and FrameNet lexical resources for estimating lexical coverage of the text on the hypothesis.

Some of these approach is closest to the method used in our present work. But, a different scoring mechanism and a different set of syntactic relations have been used in the present work. The scoring technique is quite simple and thus easy to compute and interpret.

3. SYSTEM DESCRIPTION

In this section, we describe a Lexical and Syntactic based approach for solving the Textual entailment problem. The various components of the textual entailment recognition system are pre-processing module, lexical similarity module, syntactic similarity module. Each of these modules is now being described in subsequent subsections.

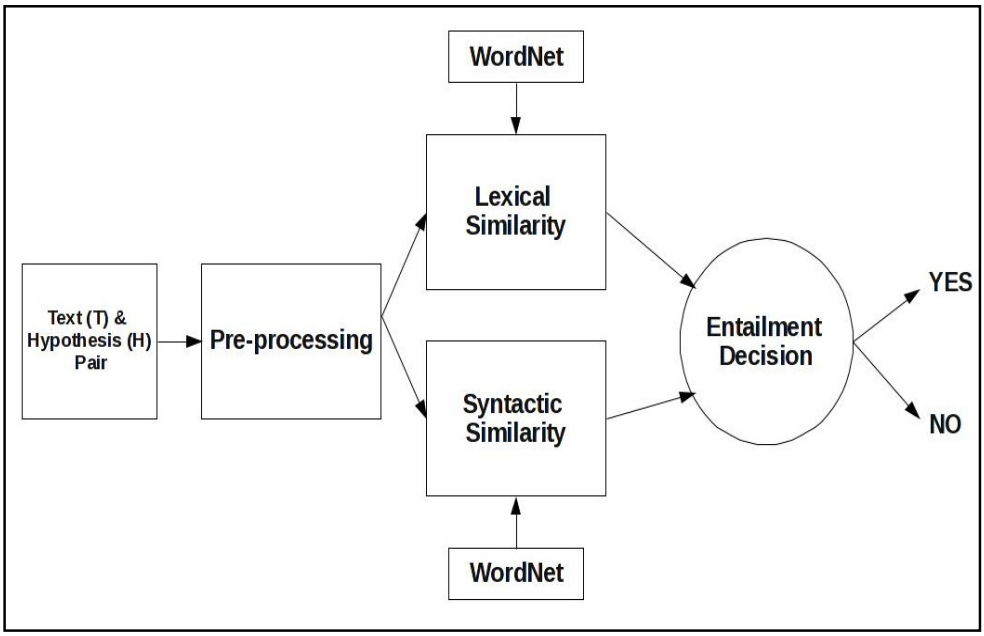

Figure 1. Textual Entailment System

3.1. Pre-processing Module

The system accepts pairs of text snippets (text and hypothesis) at the input and gives an entailment value at the output: "YES" if the text entails the hypothesis and "NO" otherwise. An example text-hypothesis pair from the set is shown in Figure 2.

RTE-2 Test Annotated Set
<**pair id**="8" **entailment**="NO" **task**="IE"> **<t>**Mangla was summoned after Madhumita's sister Nidhi Shukla, who was the first witness in the case. **</t>** **<h>**Shukla is related to Mangla. **</h>** **</pair>**
RTE-3 Development Set
<**pair id**="5" entailment="YES" task="IE" length="short" > **<t>**A bus collision with a truck in Uganda has resulted in at least 30 fatalities and has left a further 21 injured. **</t>** **<h>**30 die in a bus collision in Uganda.**</h>** **</pair>**
RTE-4 Test Set
<**pair id**="72" **entailmen**t="ENTAILMENT" **task**="IR" > **<t>**A key UN-sponsored summit has opened in Rome aimed at addressing the problem of soaring global food prices. **</t>** **<h>**UN summit targets global food crisis. **</h>** **</pair>**

Figure 2. Various RTE Data Set

The corpus has some noise as well as some special symbols that create problems during parsing. The list of such noise symbols and the special symbols is initially developed manually by looking at a number of documents and then the list is used to automatically remove such symbols from the documents. Table 1 lists the tokens that are replaced by blank as well as by other tokens. All the above pre-processing methods are applied on the development and test set as well.

Table 1. Token Replacement List

Replace by blank	Replace by Symbol	
	Original Token	Replaced Token
. –	á	a
();	č	c
[...]	è	e
()	&amp;	&
...	š	S

3.2. Lexical Similarity

In this section the various lexical features [34] 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 the text. WordNet synsets 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.

For example, let us consider the following text-hypothesis pair.

T: In the end, defeated, Antony committed suicide and so did Cleopatra, according to legend, by putting an asp to her breast.
H: Cleopatra committed suicide.

Here the common unigrams are Cleopatra, committed, suicide.

If n1= common unigram or WordNet Synonyms between text and hypothesis and n2= number of unigram in Hypothesis, then Wordnet_Unigram_Match=n1/n2.

If the value of Wordnet_Unigram_Match is 0.75 or more, i.e., 75% or more unigrams in the hypothesis match either directly or through WordNet synonyms, then the text-hypothesis pair is considered as an entailment. The text-hypothesis pair is then assigned the value of 1 meaning entailment, otherwise, the pair is assigned a value of 0.

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).

If the value of Bigram_Match is 0.5 or more, i.e., 50% or more bigrams in the hypothesis match in the corresponding text, then the text-hypothesis pair is considered as an entailment. The text-hypothesis pair is then assigned the value of 1 meaning entailment, otherwise, the pair is assigned a value of 0.

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.

If the value of LCS_Match is 0.8 or more, i.e., the length of the longest common subsequence between text T and hypothesis H is 80% or more of the length of the hypothesis, then the text-hypothesis pair is considered as an entailment. The text-hypothesis pair is then assigned the value of 1 meaning entailment, otherwise, the pair is assigned a value of 0.

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

1_skip_bigram_Match = 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.

If the value of 1_skip_bigram_Match is 0.5 or more, then the text-hypothesis pair is considered as an entailment. The text-hypothesis pair is then assigned the value of 1 meaning entailment, otherwise, the pair is assigned a value of 0.

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

If the value of Stemming_Match is 0.7 or more, i.e., 70% or more stemmed unigrams in the hypothesis match in the stemmed text, then the text-hypothesis pair is considered as an entailment. The text-hypothesis pair is assigned the value of 1 meaning entailment; otherwise, the pair is assigned a value of 0.

WordNet [36] 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 (JAWS) [37] is an API that provides Java applications with the ability to retrieve data from the WordNet database.

3.3. Syntactic Similarity

In this section the various syntactic similarity features [38] for textual entailment are described in detail. This module is based on the Stanford Parser [39], 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 Italy." 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).

Here is an example from RTE-4 data set. For the sentence, "Nigeria seizes 80 tonnes of drugs", the Stanford Dependency Parser generates the following set of dependency relations:

[
nsubj(seizes-2, Nigeria-1),
num(tonnes-4, 80-3),
dobj(seizes-2, tonnes-4),
prep_of(tonnes-4, drugs-6)
]

3.3.1. 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 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.

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.

Each of the matches through the above comparisons is assigned some weight learned from the development corpus. A threshold of 0.40 has been set on the fraction of matching hypothesis relations observed on the development set that gives optimal precision and recall values for both YES and NO entailment.

4. EXPERIMENTS AND RESULTS

We have used the following data sets: RTE-1 development set and test set, RTE-2 development set and test set, RTE-3 development set and test set, RTE-4 test set and RTE-5 main development set and test set to deal with the two-way classification task. The RTE-1 have two development sets, one consisted of 287 text-hypothesis pairs and another consisted of 287 text-hypothesis pairs. The RTE-1 test set consisted of 800 text-hypothesis pairs. Results are shown in Table 2.

Table 2. RTE-1 Development Set and Test Set Evaluation Statistics.

RTE Data	Entailment Decision	No. of Entailment in Gold standard	No. of Correct Entailment in Our System	Total No of Entailment given by our system	Precision (%)	Recall (%)	F-Score (%)
RTE-1 Development Set 1	**YES**	143	72	132	54.54	50.34	52.35
	NO	144	84	155	54.19	58.33	56.18
	Overall	287	156	287	54.35	54.35	54.35
RTE-1 Development Set 2	**YES**	140	92	176	52.27	65.71	58.22
	NO	140	56	104	53.84	40	45.89
	Overall	280	148	280	52.85	52.85	52.85
RTE-1 Test	**YES**	400	250	470	53.19	62.5	57.47
	NO	400	180	330	54.54	45	49.31
	Overall	800	430	800	53.75	53.75	53.75

The RTE-2 development set consisted of 800 text-hypothesis pairs. The RTE-2 test set consisted of 800 text-hypothesis pairs. Results are shown in Table 3.

Table 3. RTE-2 Development Set and Test Set Evaluation Statistics.

RTE Data	Entailment Decision	No. of Entailment in Gold standard	No. of Correct Entailment in Our System	Total No of Entailment given by our system	Precision (%)	Recall (%)	F-Score (%)
RTE-2 Development	**YES**	400	250	419	59.66	62.5	61.05
	NO	400	231	381	60.62	57.75	59.15
	Overall	800	481	800	60.12	60.12	60.12
RTE-2 Test	**YES**	400	272	470	57.87	68	62.52
	NO	400	202	330	61.21	50.5	55.34
	Overall	800	474	800	59.25	59.25	59.25

The RTE-3 development set consisted of 800 text-hypothesis pairs. The RTE-3 test set consisted of 800 text-hypothesis pairs. Results are shown in Table 4.

Table 4. RTE-3 Development Set and Test Set Evaluation Statistics.

RTE Data	Entailment Decision	No. of Entailment in Gold standard	No. of Correct Entailment in Our System	Total No of Entailment given by our system	Precision (%)	Recall (%)	F-Score (%)
RTE-3 Development	**YES**	412	261	373	69.97	63.34	66.49
	NO	388	276	427	64.63	71.13	67.73
	Overall	800	537	800	67.12	67.12	67.12
RTE-3 Test	**YES**	410	250	402	62.18	60.97	61.57
	NO	390	238	398	59.79	61.02	60.4
	Overall	800	488	800	61	61	61

In RTE-4 no development set was provided, as the pairs proposed were very similar to the ones contained in the RTE-3 development and test sets. Four applications, i.e., IE, IR, QA and SUM, were set as the contexts for the pair generation. The length of the H's was the same as in the past data sets (RTE-3); however, the T's were generally longer. The RTE-4 test set consisted of 1000 text-hypothesis pairs. Results are shown in Table 5.

Table 5. RTE-4 Test Set Evaluation Statistics.

RTE Data	Entailment Decision	No. of Entailment in Gold standard	No. of Correct Entailment in Our System	Total No of Entailment given by our system	Precision (%)	Recall (%)	F-Score (%)
RTE-4 Test	**YES**	500	259	464	55.81	51.8	53.73
	NO	500	295	536	55.03	59	56.94
	Overall	1000	554	1000	55.4	55.4	55.4

The RTE-5 development set consisted of 800 text-hypothesis pairs. The RTE-5 test set consisted of 800 text-hypothesis pairs. Results are shown in Table 6.

Table 6. RTE-5 Development Set and Test Set Evaluation Statistics.

RTE Data	Entailment Decision	No. of Entailment in Gold standard	No. of Correct Entailment in Our System	Total No of Entailment given by our system	Precision (%)	Recall (%)	F-Score (%)
RTE-5 Development	**YES**	300	241	414	58.21	80.33	67.5
	NO	300	127	186	68.27	42.33	52.26
	Overall	600	368	600	61.33	61.33	61.33
RTE-5 Test	**YES**	300	240	418	57.41	80	66.85
	NO	300	122	182	67.03	40.66	50.62
	Overall	600	362	600	60.33	60.33	60.33

Figure 3 shows the overall F-Score values of RTE-1 (two development sets), RTE-2, RTE-3, RTE-5 on development set. There is no development set for RTE-4.

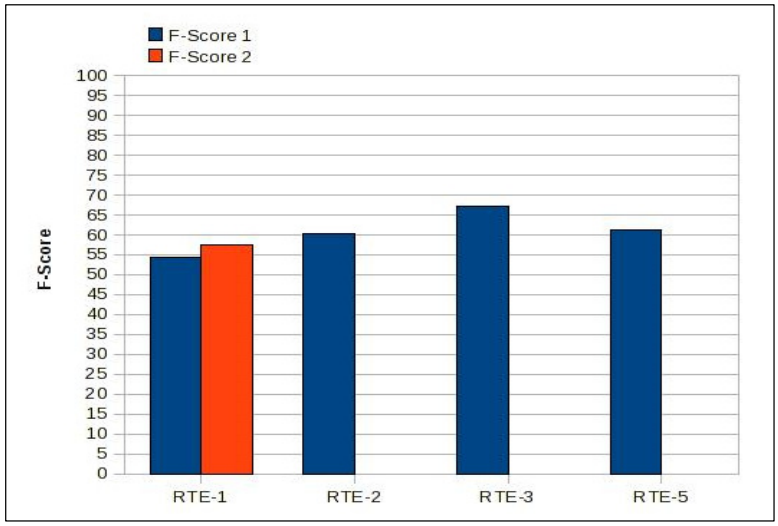

Figure 3. RTE Development Set F-Score Statistics

Figure 4 shows the overall F-Score values of RTE-1, RTE-2, RTE-3, RTE-4, RTE-5 on the respective test sets.

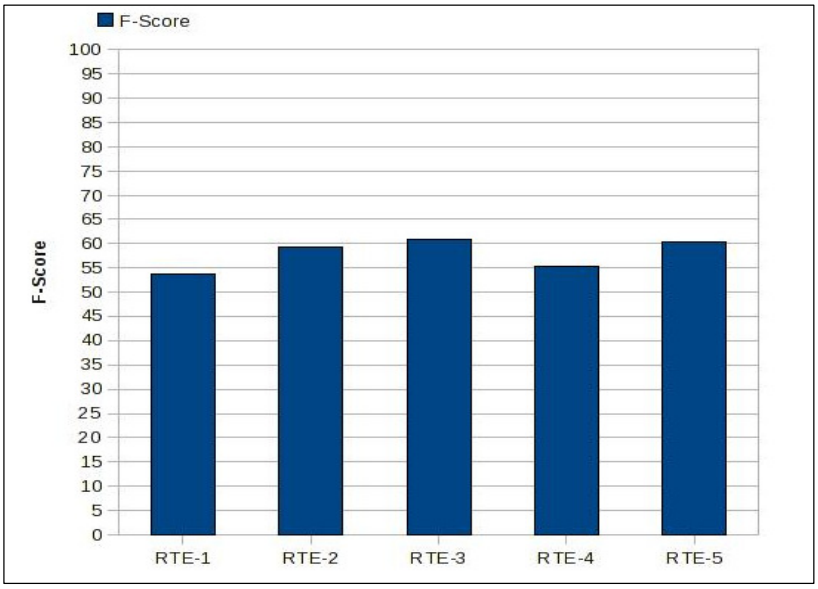

Figure 4. RTE Test Set F-Score Statistics

5. DISCUSSION

In this section we compare our results with other systems that participated in the respective RTE tracks and applied lexical and syntactic approaches. Participating system results are shown in

Table 7. The results obtained by our textual entailment system on the respective RTE tracks are shown in bold. It is observed that the results obtained by our system have outperformed the participating systems based on lexical and syntactic approaches in the respective RTE tracks.

6. CONCLUSION

Results show the effectiveness of a lexical similarity and syntactic similarity approach to handle the textual entailment problem. Experiments have been started for a semantic based RTE task. In the present task, the final RTE system has been optimized for the entailment YES/NO decision using the development set. This has to be extended for the three-way entailment decision tasks. The role of the application setting for the RTE task has not been studied in detail. This needs to be experimented in future. We want train our Textual Entailment system by lexical similarity features and syntactic features using Support Vector Machine for next set of experiments. Use of lexical information along with syntactic features and semantic features in the Textual Entailment system would be another set of interesting experiments to handle the correct decision making task.

Table 7. Compare our result with RTE participated system

Textual Entailment Challenge	System Name	Accuracy	System Description
RTE-1	Wu (HKUST)	0.512	Statistical lexical relations and Syntactic matching
	Andreevskaia (Concordia)	0.519	WordNet and Syntactic matching
	Zanzotto (Rome-Milan)	0.524	WordNet and Syntactic matching
	Our TE System	**0.537**	**Lexical Relation, WordNet and Syntactic matching**
RTE-2	Herrea (UNED)	0.588	Lexical, Syntactic Matching, ML Classification
	Kozareva (Alicante)	0.558	Lexical, Syntactic Matching, Corpus/ Web based statistics, ML Classification
	Inkpen (Ottawa)	0.581	Lexical, n-gram overlap, Syntactic Matching, ML Classification
	Our TE System	**0.592**	**Lexical Relation, WordNet and Syntactic matching**
RTE-3	Harmling	0.5775	Lexical Relation, WordNet, Syntactic Matching\Aligning, ML Classification
	Blake	0.6050	Lexical Relation, WordNet, Syntactic Matching/Aligning, ML Classification
	Ferrés	0.6062	Lexical Relation, WordNet, Syntactic Matching/Aligning, ML Classification
	Our TE System	**0.61**	**Lexical Relation, WordNet and Syntactic matching**
RTE-4	Emory3	0.511	Lexical, Syntactic and Semantic Relation
	Yatbaz	0.509	Dependency tree
	FSC	0.526	Word Similarity
	Our TE System	0.554	Lexical Relation, WordNet and Syntactic matching
RTE-5	AUEBNLP	0.556	Lexical, Syntactic Relation
	JU_CSE_TAC	0.55	Lexical Relation
	UB.dmirg1	0.50	Lexical, Syntactic Relation
	Our TE System	**0.603**	**Lexical Relation, WordNet and Syntactic matching**

REFERENCES

[1] Dagan, I., Glickman, O., Magnini, B.: The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[2] Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B., Szpektor, I.: The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy (2006).

[3] 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, Prague, Czech Republic. (2007).

[4] Giampiccolo, D., Dang, H. T., Magnini, B., Dagan, I., Cabrio, E.:The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings. http://www.nist.gov/tac/publications/2008/papers.html (2008)

[5] Bentivogli, L., Dagan, I., Dang. H.T., Giampiccolo, D., Magnini, B.: The Fifth PASCAL Recognizing Textual Entailment Challenge, In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA. (2009).

[6] Luisa Bentivogli, Peter Clark, Ido Dagan, Hoa Trang Dang,Danilo Giampiccolo: The Sixth PASCAL Recognizing Textual Entailment Challenge. In TAC 2010 Notebook Proceedings. (2010)

[7] Yuret, D., Han, A., Turgut, Z., SemEval-2010 Task 12: Parser Evaluation using Textual Entailments, Proceedings of the SemEval-2010 Evaluation Exercises on Semantic Evaluation. (2010).

[8] Voorhees, E.M., Harman, D.: Overview of the seventh text retrieval conference. In Proceedings of the Seventh Text REtrieval Conference (TREC-7). NIST Special Publication. (1999).

[9] Dekai Wu, "Textual Entailment Recognition Based on Inversion Transduction Grammars", The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[10] Alina Andreevskaia, Zhuoyan Li and Sabine Bergler, "Can Shallow Predicate Argument Structures Determine Entailment?", The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[11] Maria Teresa Pazienza, Marco Pennacchiotti, Fabio Massimo Zanzotto, "Textual Entailment as Syntactic Graph Distance: a rule based and a SVM based approach", The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[12] Johan Bos, Katja Markert. "Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment", The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[13] Jesus Herrera, Anselmo Penas, Felisa Verdejo. "Textual Entailment Recognition Based on Dependency Analysis and WordNet", The PASCAL Recognising Textual Entailment Challenge Proceedings of the First PASCAL Recognizing Textual Entailment Workshop. (2005).

[14] Jesus Herrera, Anselmo Penas, Alvaro Rodrigo, Felisa Verdejo, "UNED at PASCAL RTE-2 Challenge", The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy. (2006).

[15] Zornitsa Kozareva Andres Montoyo, "MLEnt: The Machine Learning Entailment System of the University of Alicante", The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy. (2006).

[16] Diana Inkpen, Darren Kipp, and Vivi Nastase, "Machine Learning Experiments for Textual Entailment", The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy. (2006).

[17] Frank Schilder, Bridget Thomson McInnes: "Word and tree-based similarities for textual entailment", The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy. (2006).

[18] Milen Kouylekov and Bernardo Magnini, "Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion", The Second PASCAL Recognising Textual Entailment Challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venice, Italy. (2006).

[19] Adams, R.: Textual Entailment Through Extended Lexical Overlap. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, pp. 128-133, 2006.

[20] Stefan Harmeling, "An Extensible Probabilistic Transformation-based Approach to the Third Recognizing Textual Entailment Challenge", The Third PASCAL Recognizing Textual Entailment Challenge, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic. (2007).

[21] Catherine Blake, "The Role of Sentence Structure in Recognizing Textual Entailment", The Third PASCAL Recognizing Textual Entailment Challenge, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic. (2007).

[22] Daniel Ferres, Horacio Rodriguez, "Machine Learning with Semantic-Based Distances Between Sentences for Textual Entailment", The Third PASCAL Recognizing Textual Entailment Challenge, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic. (2007).

[23] Adrian Iftene, Alexandra Balahur-Dobrescu, "Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment", The Third PASCAL Recognizing Textual Entailment Challenge, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic. (2007).

[24] Blake, C.: The Role of Sentence Structure in Recognizing Textual Entailment. Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp101-106, 2007.

[25] Wang, R., Neumann, G.: Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons. Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 36-41, 2007.

[26] Malakasiotis, P., Androutsopoulos, I.: Learning Textual Entailment using SVMs and String Similarity Measures, Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 42-47 , 2007.

[27] Eugene Agichtein, Walt Askew, Yandong Liu, "Combining Lexical, Syntactic, and Semantic Evidence For Textual Entailment Classification", The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings. http://www.nist.gov/tac/publications/2008/papers.html. (2008).

[28] Mehmet Ali Yatbaz, "RTE4: Normalized Dependency Tree Alignment Using Unsupervised N-gram Word Similarity Score", The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings. http://www.nist.gov/tac/publications/2008/papers.html. (2008).

[29] Orlando Montalvo-Huhn, Stephen Taylor, "Textual Entailment – Fitchburg State College", The Fourth PASCAL Recognizing Textual Entailment Challenge. In TAC 2008 Proceedings. http://www.nist.gov/tac/publications/2008/papers.html. (2008).

[30] Roy Bar-Haim, Jonathan Berant, Ido Dagan, Iddo Greental, Shachar Mirkin, Eyal Shnarch and Idan Szpektor, "Efficient Semantic Deduction and Approximate Matching over Compact Parse

Forests", In TAC 2008 Proceedings. http://www.nist.gov/tac/publications/2008/papers.html. (2008).

[31] Varma, V., Krishna, S., Garapati, H., Reddy, K., Pingali, P., Ganesh, S., Gopisetty, H., Bysani, P., Katragadda, R., Sarvabhotla, K., Reddy, V.B.,Bharadwaj,R.: Recognizing Textual Entailment (RTE) Track. In Text analysis conference 2008 Proceedings, 2008.

[32] Castillo, J.J., Alemany, L.A.: An approach using Named Entities for Recognizing Textual Entailment, Recognizing Textual Entailment (RTE) Track. In Text analysis conference 2008 Proceedings. (2008).

[33] Prodromos Malakasiotis, "AUEB at TAC 2009", In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA, 2009.

[34] Partha Pakray, Sivaji Bandyopadhyay, Alexander Gelbukh, "Lexical based two-way RTE System at RTE-5", In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA, 2009.

[35] Bahadorreza Ofoghi, John Yearwood, "UB.dmirg: A Syntactic Lexical System for Recognizing Textual Entailments", In TAC 2009 Workshop, National Institute of Standards and Technology Gaithersburg, Maryland USA, 2009.

[36] Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998).

[37] Java API for WordNet Searching, Available: http://lyle.smu.edu/~tspell/jaws/index.html.

[38] Partha Pakray, Alexander Gelbukh and Sivaji Bandyopadhyay, "A Syntactic Textual Entailment System Using Dependency Parser", Springer Berlin / Heidelberg, Volume Volume 6008/2010, Book Computational Linguistics and Intelligent Text Processing, ISBN 978-3-642-12115-9, Pages 269-278.

[38] Stanford Parser Tools, Available: http://nlp.stanford.edu/software/lex-parser.shtml.