Sagan in TAC2010: A Machine Learning Approach to RTE within a Corpus

Julio Javier Castillo

Faculty of Mathematic Astronomy and Physics - National University of Cordoba

Regional Faculty of Cordoba – National Technological University
Córdoba, Argentina
jotacastillo@gmail.com

Abstract

This paper describes the Sagan system in the context of the Sixth Pascal Recognizing Textual Entailment (RTE6) and the RTE task within a Corpus.

The system employs a Support Vector Machine classifier which uses semantic similarity metrics to sentence level using only WordNet as source of knowledge, and co-reference analysis. Additionally, we proposed a baseline to the Novelty Detection subtask.

Keywords

Textual entailment search task, Textual entailment within Corpus, recognizing textual entailment, machine learning.

1. Introduction

The goal of the RTE Track is to develop systems that recognize when one piece of text (T) entails another (H).

This year the National Institute of Standards and Technology (NIST) organized the Text Analysis Conference (TAC) 2010, which has three main tracks, namely Knowledge Base Population (KBP), Recognizing Textual Entailment (RTE), and Summarization, providing a common evaluation framework of different NLP tasks.

In order to move the RTE task towards more realistic application scenarios the texts come from a variety of sources and may include typographical errors and ungrammatical sentences.

The RTE6 Main data set is based on the data created for the TAC 2009 Update Summarization task, consisting of a number of topics, each containing two sets of documents, namely "Cluster A", composed of the first 10 texts in chronological order of publication date, and "Cluster B", composed of the last 10 texts.

The systems must find all the entailing sentences (Ts) in a corpus of 10 newswire documents about a common topic, previously filtered by Lucene[1].

So, the main difference with respect to the main task is that in the Entailment Search task both Text and Hypothesis are to be interpreted in the context of the corpus.

The texts T's (candidate entailing sentences) are the 100 top-ranked sentences retrieved for each text H by Lucene taken from the "Cluster A" corpus.

Thus, this task consists of finding all the sentences in a set of documents that entails a given Hypothesis such as is defined on [1].

In this paper, we present the Sagan system to address the textual entailment recognition task within a Corpus. The system applies a Support Vector Machine classifier to the problem of recognizing textual entailment search task. So, the SVM classify the candidate RTE6 test pairs in two classes: Entailment (Yes), No Entailment (NO).

This year, we modify our past Sagan system [2] in order to work almost exclusively with semantic features, with the aims of exploring more deeply how semantic information could help in the RTE task, specially the benefit of WordNet as knowledge resource. Then, we use 8 selected WordNet-based features. These features are used to characterize the relationship between text and hypothesis for both training and test cases.

We also propose a baseline to the Novelty Detection subtask. This task has the same structure as the Main Task, with the objective of judging whether the information contained in each H is novel with respect to the information contained in a given corpus. A text fragment H is defined as "new" whether there is no entailing sentences in the set of candidate text T's.

The remainder of the paper is organized as follows: Section 2 describes the architecture of our system, whereas Section 3 shows the experimental evaluation and discussion of results.

Finally, Section 4 summarizes the conclusions and lines for future work.

[1] http:// lucene.apache.org/

2. System Architecture

This section provides an overview of our system as used for RTE6 track at the TAC 2010 Challenge. The system is based on a machine learning approach for recognizing textual entailment.

We use a supervised machine learning approach to train a SVM classifier over a set of WordNet-based semantic metrics.

The system produces feature vectors for the RTE 3 dataset (previously converted in a two-way task). Additionally, we use the following training sets: RTE3-4C[2], and RTE4-4C[2] such as we described in [3] in order to extend the RTE data sets by using machine translation engines.

The augmented corpus is denoted RTE3-4C and is composed of: 340 pairs Contradiction, 1520 pairs Entailment, and 1114 pairs Unknown. So, for the two-way task it is composed of: 1454 pairs No(No Entailment), and 1520 pairs Yes(Entailment).

In the case of the RTE4-4C data set, it has the following composition: 546 pairs Contradiction, 1812 pairs Entailment, and 1272 pairs Unknown. Therefore, in the two-way task, there are 1818 pairs No, and 1812 pairs Yes, in this data set.

We submitted three runs, and each training set was used in a different run.

Thus, we used eight WordNet-based measures with the aim of obtain the maximum similarity between two concepts. The measures used are: Resnik [4], Lin [5], Jiang & Conrath [6], Pirro & Seco [7], Wu & Palmer [8], Path Metric, Leacock & Chodorow [9], and a semantic similarity to sentence level [10], which we named SenSim in this paper.

The motivation of these input features was to test our system over a wide range of semantic feature and try to determinate the accuracy obtained only with the semantic information provided by WordNet. Further improvement on the system can be done using lexical and syntactic features.

We tried to model the semantic similarity of two texts (T,H) as a function of the semantic similarity of the constituent words of both phrases. In order to reach this objective, we used a text to text similarity measure which is based on word to word similarity. We expect that combining word to word similarity metrics to text level would be a good indicator of text to text similarity.

Figure 1 shows the overview architecture of the System.

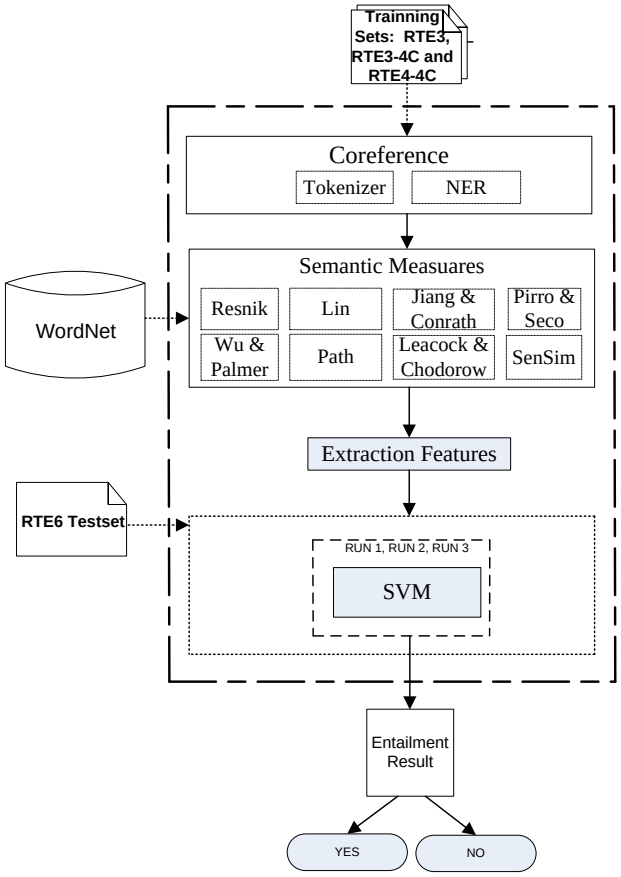

Figure 1.General architecture of the system

First, the <T,H> pairs are pre-processed with a coreference analysis.

Second, we compute eight WordNet-based semantic features.

Finally, for every submitted runs we use a SVM classifier for the RTE search classification task reducing the corpus to a set of Text and Hypothesis pairs and then applying the traditional techniques for RTE over each pair.

2.1 Preprocessing

The Preprocessing module has two sub-modules:

Tokenizer: The text-hypothesis pairs are tokenized with the tokenizer of OpenNLP[3] framework.

NER: Name Entity Recognition Tool of OpenNLP[3] framework.

Three runs were submitted to the Textual Analysis Conference 2010 differing only in the training set used. For RUN1 we use 800 pairs of the RTE3 devset, 2974 pairs of

[2] http://www.investigacion.frc.utn.edu.ar/mslabs/~jcastillo/Sagan-test-suite/

[3] http://opennlp.sourceforge.net/

RTE3-4C, and 3630 pairs of RTE4-4C, for RUN2 and RUN3 respectively.

In order to deal with the coreference issues, we propose the following algorithm which works to document level resolutions:

1. For each $Topic_k$, document Dj.
2. Appends a Hypothesis Hi to the end of the document Dj.
3. Computes a coreference analysis over the document Dj.
4. Identifies all coreferences that refer to the same entity.
5. Takes the longest reference which refers to the same entity.
6. Replaces within the document Dj the reference obtained on the step 5 (in the places identified on the step 4).

The following example presents as an entity will be replaced by an equivalent entity adding redundant information. For example, running a NER tool from OpenNLP and using the above algorithm, we obtained a set of pairs ["noun phrase", noun phrase id], below we show the first two:

["Bobby Fischer is a chess master", 18]

["Fischer, the 62-year-old chess champion, is a chess master", 18]

The first piece of text represents the noun phrase that is being referenced and the second number is a unique reference id of the noun phrase in the whole document.

Thus, the algorithm selects "Fischer, the 62-year-old chess champion, is a chess master" and replaces all references with the same id, using this noun phrase.

By performing the previous procedure there is likely to obtain some text syntactically incorrect, but still understandable. We expect that the meaning of the sentence (text fragment) remains the same.

In the case that equal length names are found, then the algorithm chooses the first occurrence of the entity with maximum length.

From a semantic point of view, the H-modified, generally provides more information than the original hypothesis H.

Once this process is performed, every <T,H> candidate pair (previously filtered with Lucene) of a document is taken and fed into the system, as explained before, following the preprocessing procedure with outputs True/False.

Preliminary experiments suggest that changing a noun phrase with their longest occurrence helps to capture more lexical similarity overlaps that could also provide higher scores in the semantics feature.

2.2 Features

WordNet, the most popular source of knowledge in RTE, is used to calculate the semantic similarity between a T (Text) and an H (Hypothesis). The following procedure is applied:

2.2.1. Word-Level Features

Step 1. Perform WSD (Word Sense Disambiguation) using the Lesk algorithm [11], based on WordNet definitions (glosses).

Step 2. A semantic similarity matrix between words in T and H is defined.

Step 3. A function Fsim is applied to *T* and *H*.

Where the Function Fsim could be one of the followings eight functions (seven Function plus SemSim function, see section 2.2.2) over concepts *s*, and *t*:

Function 1. The Resnik [4] similarity metric measures the information content (IC) of the two WordNet concepts *s* and *t* by using LCS:

$$RES(s,t) = IC(LCS(s,t))$$

And IC is defined as:

$$IC(w) = -\log P(w)$$

Where: *P(w)* is the probability of finding the word *w* in a large corpus in English, and LCS(s,t):is the least common subsume of *s* and *t*.

Function 2. The Lin [5] similarity metric, is based on Resnik's measure of similarity, and adds a normalization factor consisting of the information content of the two input concepts *s* and *t*:

$$LIN(s,t) = \frac{2 * IC(LCS(s,t))}{IC(s,t)}$$

Function 3. Another metric considered is Jiang & Conrath [6] which is defined as:

$$JICO(s,t) = \frac{1}{IC(s) + IC(t) - 2 * IC(LCS(s,t))}$$

The word similarity measures are normalized in [0–1].

The normalization is done by dividing the similarity score provided by the maximum score of that measure.

Function 4. The Pirro & Seco [7] similarity metric is also based on Resnik's measure of similarity, but is defined by using information theory and solving a problem with Resnisk's measure when computing the similarity between identical concepts yielding the information content value of their most specific common abstraction that subsumes two different concepts (*msca*). In the practice, *msca* gives the most specific common abstraction value for the two given synsets, where the synsets are represented as Lucene

documents. So, the Pirro & Seco (*PISE*) similarity metric is the following:

$$PISE(s,t) = \begin{cases} 3 * IC(msca(s,t)) - IC(s) - IC(t), if\ s \neq t \\ 1 \qquad\qquad\qquad\qquad\quad ,if\ s = t \end{cases}$$

Function 5. The Wu & Palmer [13] measure is based on path length between concepts:

$$WUPA(C_1(s), C_2(t)) = \frac{2 * N_3}{N_1 + N_2 + 2 * N_3}$$

Where: *C1* and *C2* are the synsets to which *s* and *t* belongs, respectively. *C3* is the least common superconcept of *C1* and *C2*. *N1* is the number of nodes of the path from *C1* to *C3*. *N2* is the number of nodes of the path from *C2* to *C3*. *N3* is the number of nodes on the path from *C3* to root.

Function 6. The Path metric is the reciprocal of the length of the shortest path between 2 synsets. Note that we count the 'nodes' (synsets) in the path, not the links. The allowed POS types are nouns and verbs.

It is an easy and fast method of getting similarity applying a notion of 'semantic relatedness' via node counting, and is defined as:

$$PA(s,t) = Min_i(PathLength_i(s,t))$$

Where: $PathLength_i(s,t)$ gives the length of the i-Path between *s* and *t*.

Function 7. The Leacock & Chodorow [9] metric finds the path length between *s* and *t* in the "is-a" hierarchy of WordNet. In order to obtain the relatedness of the two concepts, the path length is divided by the depth of the hierarchy (D) in which they reside. Our implementation applies the basic version of this measure by using "fake roots".

$$LECH(C_1(s), C_2(t)) = -\log\left(\frac{Min_i(PathLength_i(s,t)}{2 * D}\right)$$

Where: *D* = is the maximum depth of the taxonomy (considering only nouns and verbs).

Step 4. Finally, the string similarity between two lists of words is reduced to the problem of bipartite graph matching, being performed by using the Hungarian algorithm over this bipartite graph. Then, we find the assignment that maximizes the sum of ratings of each token. Note that each graph node is a token/word of the list.

At the end, the final score is calculated by:

$$finalscore = \frac{\sum_{s \in T, t \in H} opt(Fsim(s,t))}{Max\ (Length(T), Length\ (H))}$$

Where: opt(*F*) is the optimal assignment in the graph.

Length (*T*) is the number of tokens in *T*, Length (*H*) is the number of tokens in *H*, and

$$Fsim \in \{RES, LIN, JICO, PISE, WUPA, PA, LECH\}.$$

Finally, note that the partial influence on each of the individual similarity is going to be reflected on the overall similarity.

2.2.2. SemSim: WordNet-based sentence similarity metric

To build our feature vector we use an additional metric such as [10] to compute sentence level similarity.

This metric that we called "SemSim" (Function 8) is used to calculate the semantic similarity between a *T* and a *H*. The following procedure is applied:

Step 1. Word sense disambiguation using the Lesk algorithm, based on the definitions of WordNet.

Step 2. A semantic similarity matrix between words in *T* and *H* is defined. Words are used only in synonym and hyperonym relationship. The Breadth First Search algorithm is used over these tokens; similarity is calculated using two factors: length of the path and orientation of the path. The semantic similarity between two words or concepts *s* and *t*, is computed as:

$$Sim(s,t) = 2 \times \frac{Depth(LCS(s,t))}{Depth(s) + Depth(t)}$$

Where: *Depth(s)* is the shortest distance from the root node to the current node.

Step 3. To obtain the final score, matching average between two sentences *T* and *H* is calculated as follows:

$$SemSim(T,H) =$$

$$MatchingAverage(T,H) = 2 \times \frac{Match\ (T,H)}{Length\ (T) + Length\ (H)}$$

2.3 Textual Entailment Novelty Detection Subtask

The Novelty Detection subtask is based on the Main Task focusing on detection of novelty in Cluster B documents.

The task consists of judging whether the information contained in each text H is novel with respect to the information contained in the set of Cluster A candidate entailing sentences T's.

When no entailing sentences are detected, then the information contained in the H is novel.

On the other hand, if one or more entailing sentences are found, then the content of the H is not new.

Thus, we proposed a baseline based on Name Entities to the novelty detection subtask, and the algorithm proposed is as follow:

1) Build a bag of word W with all name entities of the candidate sentences T's.
2) Extend this bag W with acronyms of these entities.

3) Determine whether exists a name entity in H which is not present in W.

If entailment sentences are found for a given H, it means that the context of the H is not new, and if no entailment sentences are detected, it means that information contained in the H is novel.

F1	***Score***
High	0.8291
Median	0.7784
Baseline(RUN1)	0.4398

Table 1.Results of the baseline proposed

This baseline is easy to implement and is based exclusively on Name Entities and their variations (acronyms). We see that others teams have overcome the baseline proposed.

3. Experimental Evaluation and Discussion of Results

Eighteen teams submitted a total of 48 run to the "RTE within a Corpus" task. In this context, our RUNs are clearly above the system with low score, but are below the average score system. Our official results for RTE6 test set are summarized in Table 2.

F1	***Score***
High	0.4801
Median	0.3372
RTE3(RUN1)	0.2409
RTE3-4C(RUN2)	0.22.29
RTE4-4C(RUN3)	0.2019
Low	0.1160

Table 2.Results obtained with different training set in order to predict RTE6.

Table 2 shows the three runs submitted to Textual Analysis Conference 2010, and also shows the high score and low score of the RTE6 participants.

In [3] we showed that by using RTE3-4C is possible to improve the accuracy in comparison with RTE3 in the "traditional" RTE task. Curiously, Table 2 shows that RTE3 alone present better results that RTE3-4C. Although the difference is not statistically significant is interesting to note that RTE3-4C has three times more pairs than RTE3. RTE3-4C was generated using machine translation which expanded the RTE3 corpus enriching the lexical and semantic variability.

So, one reason for this discrepancy could be that it is necessary to increase the number of features, and the characteristic of the RTE6 (which is based on SUM subtask) and is a real text corpus. Furthermore, it was set up in the Summarization setting, attempting to analyze the potential impact of textual entailment on a real NLP application.

The best performance of our system was achieved with RUN1, and it was 24.09% acc, then RUN2 with 22.29% of accuracy, and finally the RUN3 with 20.19%.

The accuracy of RUN1 is placed 9% below of median score, and 13% above the low score, showing that further improvement are needed.

Thus, we conclude that this semantic approach is very preliminary mainly due we are using and exploiting only one resource: WordNet. Surely, by using additional semantic resources we could improve the accuracy of our system.

This year, again ablation tests are mandatory for systems participating in Textual Analysis Conference 2010, with the aims of collecting data to better understand the impact of the knowledge resources used by RTE systems and evaluate the contribution of each resource to system performance. It is implemented removing one of the resources of the system. However, since our system is based on only one resource, the ablation test has not sense in our system.

Table 3 shows the results obtained results by topic for the main task for RUN1. The difference in F-measure between the high and low scores is 0.11, showing that indeed some topics seem to be easier than others when predicting using WordNet as the only resource.

Topic	***Precision***	***Recall***	***F-measure***
"D0901"	0.1467	0.4118	0.2163
"D0902"	0.1869	0.3978	0.2543
"D0907"	0.1757	0.7027	0.2811
"D0913"	0.1355	0.6463	0.2241
"D0918"	0.1293	0.4787	0.2036
"D0928"	0.1943	0.4141	0.2645
"D0931"	0.1992	0.4766	0.2810
"D0936"	0.1972	0.6176	**0.2989**
"D0939"	0.1250	0.3391	0.1827
"D0943"	0.1646	0.6031	0.2586

Table 3. Results of Sagan system for RUN1 divided by topic.

In this approach we tried to test only the benefit of the WordNet semantic resource, and therefore we chose a representative set of WordNet-based semantic measure, which was previously extended to work at sentence level.

Despite of the fact that our very simple approach we think that a lot of improvements could be done in order to increase the F-score of the Sagan system, mainly adding more knowledge resources and features refining the before algorithm.

4. Conclusion and Future Work

In this paper we show the Sagan system approach which uses a set of semantic features that uses only WordNet as semantic resource to try to determine how semantic information helps in the textual entailment semantic task.

We described our submission for the Recognizing Textual Entailment main track, and we also report our participation in the Textual Entailment Novelty Detection Subtask.

As conclusion, we need to explore the improvement that can be achieved with a combination of a rich set of lexical, syntactic and semantic measures based on a spectrum of knowledge resources.

On the other hand, our approach to Textual Entailment Text within a Corpus is focused on quantify the improvement of the most common used knowledge resource in RTE, which is WordNet.

Future work is oriented to experiment with additional lexical, syntactic and semantic similarities features and test the improvements they may yield among a wide spectrum of knowledge resources.

5. References

[1] L. Bentivogli, I. Dagan, H. T. Dang, D. Giampiccolo, and B. Magnini, *The Fifth PASCAL Recognizing Textual Entailment Challenge.* In proceedings of Textual Analysis Conference. NIST, Maryland USA, 2009.

[2] J. Castillo.: *Textual Entailment Search Task: An Initial Approach Based on Coreference Resolution*. In ICICCI '10 Proceedings of the 2010 International Conference on Intelligent Computing and Cognitive Informatics ICICCI2010,IEEE- Xplorer. 2010.

[3] J. Castillo.: *Using Machine Translation Systems to Expand a Corpus in Textual Entailment*. In Lecture Notes in Computer Science, volume 6233, pages 97-102.

[4] Resnik, P.: *Information Content to Evaluate Semantic Similarity in a Taxonomy*. In Proc. Of IJCAI 1995, pp. 448-453 (1995).

[5] Lin, D.: *An Information-Theoretic Definition of Similarity*. In Proc. of Conf. on Machine Learning, pp. 296-304 (1998).

[6] Jiang,J., Conrath, D.: *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*. In Proc. ROCLING X (1997).

[7] Pirrò, G., Seco, N.: *Design, Implementation and Evaluation of a New Similarity Metric Combining Feature and Intrinsic Information Content*. ODBASE 2008, LNCS (2008).

[8] Wu, Z., Palmer, M.: *Verb semantics and lexical selection*. In 32nd ACL (1994).

[9] Leacock, C., Chodorow, M.: *Combining local context and WordNet similarity for word sense identification*. MIT Press. 265–283 (1998).

[10] J.Castillo.: *A Semantic Oriented Approach to Textual Entailment using WordNet-based measures*. In Lecture Notes in Computer Science, Volume 6437, pages 44-55.

[11] M. Lesk. *Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone.* In SIGDOC '86, 1986.

[12] Gusfield, Dan. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* CUP, 1999.

[13] Bill Dolan, Chris Quirk, and Chris Brockett. 2004. *Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources.* In COLING '04: Proceedings of the 20th international conference on Computational Linguistics, page 350, Morristown, NJ, USA. Association for Computational Linguistics.

[14] Marie-Catherine de Marneffe, et al. Manning.*Learning to distinguish valid textual entailments*. In Proceedings of the Third Recognizing Textual Entailment Challenge, Italy, 2006.

[15] M. Lesk. *Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone.* In SIGDOC '86, 1986.

[16] Gusfield, Dan. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology.* CUP, 1999.

[17] J. Castillo, M. Cardenas.: *Using Sentence Semantic Similarity Based on WordNet in Recognizing Textual Entailment*. 12th Ibero-American Conference on AI, IBERAMIA 2010, In Lecture Notes in Computer Science, Volume 6433, pages 366-375.

[18] L. Bentivogli, P. Clark, I. Dagan, H. T. Dang, and D. Giampiccolo.: *The Sixth Pascal Recognizing Textual Entailment Challenge*. In proceedings of Textual Analysis Conference. NIST, Maryland USA, 2010.