A Machine Learning Approach for Recognizing Textual Entailment in Spanish

Julio Javier Castillo
National University of Córdoba
Ciudad Universitaria, 5000
Córdoba, Argentina
jotacastillo@gmail.com

Abstract

This paper presents a system that uses machine learning algorithms for the task of recognizing textual entailment in Spanish language. The datasets used include SPARTE Corpus and a translated version to Spanish of RTE3, RTE4 and RTE5 datasets. The features chosen quantify lexical, syntactic and semantic level matching between text and hypothesis sentences. We analyze how the different sizes of datasets and classifiers could impact on the final overall performance of the RTE classification of two-way task in Spanish. The RTE system yields 60.83% of accuracy and a competitive result of 66.50% of accuracy is reported by train and test set taken from SPARTE Corpus with 70% split.

1 Introduction

The objective of the Recognizing Textual Entailment Challenge is determining whether the meaning of the Hypothesis (H) can be inferred from a text (T) (Ido Dagan et al., 2006). This challenge has been organized by NIST in recent years.

Another related antecedent was Answer Validation Exercise (AVE), part of Cross Language Evaluation Forum (CLEF), whose objective is to develop systems which are able to decide whether the answer to a question is correct or not (Peñas et al, 2006). It was a three year-old track, from 2006 to 2008.

AVE challenge was an evaluation framework for Question Answering (QA) systems to promote the development and evaluation of subsystems aimed at validating the correctness of the answers given by a QA system. The Answer Validation task must select the best answer for the final output. There is a subtask for each language involved in QA, the Spanish is one of these. Thus, AVE task is very similar to RTE (Recognition of Textual Entailments).

In this paper, we address the RTE task problem of determining the entailment value between Text and Hypothesis pairs in Spanish, applying machine learning techniques.

In the past, RTEs Challenges machine learning algorithms were widely used for the task of recognizing textual entailment (Marneffe et al., 2006; Zanzotto et al., 2007; Castillo, 2009) and they have reported goods results for English language. Also, our system applies machine learning algorithms to the Spanish.

We built a set of datasets based on public available datasets for English, together to SPARTE (Peñas et al, 2006), an available Corpus in Spanish. This corpus contains 2962 hypothesis with a document label and a True/False value indicating whether the document entails the hypothesis or not. Up to our knowledge, SPARTE corpus in the only corpus aimed at evaluating RTE systems in Spanish.

Finally, we generated a feature vector with the following components for both Text and Hypothesis: Levenshtein distance, a lexical distance based on Levenshtein, a semantic similarity measure Wordnet based, and the LCS (longest common

Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas, pages 62–67, Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

substring) metric; in order to characterize the relationships between the Text and the Hypothesis.

The remainder of the paper is organized as follows. Section 2 shows the system description, whereas Section 3 describes the results of experimental evaluation and discussion of them. Section 4 discusses opportunities of collaboration. Finally, Section 5 summarizes the conclusions and lines for future work.

2 System Description

This section provides an overview of our system which is based on a machine learning approach for recognizing textual entailment to the Spanish. The system produces feature vectors for the available development data RTE3, RTE4, RTE5, and SPARTE(Peñas et al, 2006). Weka (Witten and Frank, 2000) is used to train classifiers on these feature vectors.

The SPARTE Corpus, was built from the Spanish corpora used at Cross-Language Evaluation Forum (CLEF) for evaluating QA systems during the years 2003, 2004 and 2005. This corpus contains 2962 hypothesis with a True/False value indicating whether the document entails the hypothesis or not.

Due to, all available dataset of PASCAL Text Analysis Conference were in English, we translated every dataset to Spanish by using an online translator engine[1]. So, we had a Spanish dataset but with some translation errors provided by the translator. It is important to note, that the "quality" of the translation is given by the Translator engine, and we suppose that the sense of the sentence should not be modified by the Translator. Indeed, it is the situation for the majority of the cases that we analyzed. The new datasets were named RTE3-Sp (Spanish), RTE4-Sp, and RTE5-Sp.

The following example is the pair number 799 from RTE3-Sp with False as entailment value.

Text:

Otros dos marines, Tyler Jackson y Juan Jodka III, ya han se declaró culpables de asalto agravantes y conspiración para obstruir la justicia y fueron condenados a 21 meses y 18 meses, respectivamente.

Hypothesis:

Tyler Jackson ha sido condenado a 18 meses.

This example shows a little noisy (and a minimal syntactic error) in the translation of the Text to Spanish (instead of "*ya han se declaró*" should be "ya se han declarado"); but the whole meaning was not changed.

Also, we show a pair example (pair id=3) taken from Sparte Corpus with False as entailment value:

Text: *¿Cuál es la capital de Croacia?*

Hypothesis :

La capital de Croacia es ONU.

In a similar way, all pairs from SPARTE belong to QA task and these are syntactically simpler than RTE's Corpus pairs.

Additionally, we generate the following development sets: RTE3-Sp+RTE4-Sp, and SPARTE-Bal+RTE3-Sp+RTE4-Sp in order to train with different corpus and different sizes. In all cases, RTE5-Sp TAC 2009 gold standard dataset was used as test-set.

Also, we did additional experiments with SPARTE, using cross-validation technique and percentage split method, in order to test the accuracy of our system taking only this corpus as development and training set.

2.1 Features

We experimented with the following four machine learning algorithms: Support Vector Machine (SVM), Multilayer Perceptron(MLP), Decision Trees(DT) and AdaBoost(AB).

The Decision Trees are interesting because we can see what features were selected from the top levels of the trees. SVM and AdaBoost were selected because they are known for achieving high performances, and MLP was used because it has achieved high performance in others NLP tasks.

We experimented with various settings for the machine learning algorithms, including only the results for the best parameters.

We generated a feature vector with the following components for every possible <T,H>: Levenshtein distance, a lexical distance based on Levenshtein, a semantic similarity measure Word-

[1] http://www.microsofttranslator.com/

net based, and the LCS (longest common substring) metric.

We chose only four features in order to learn the development sets, having into account that larger feature sets do not necessarily lead to improving classification performance because it could increase the risk of overfitting the training data.

Below the motivation for the input features: Levenshtein distance is motivated by the good results obtained as a measure of similarity between two strings. Using stems, this measure improves the Levenshtein over words. The lexical distance feature based on Levenshtein distance is interesting because works to a sentence level. Semantic similarity using WordNet is interesting because of the capture of the semantic similarity between T and H to sentence level. Longest common substring is selected because it is easy to implement and provides a good measure for word overlap.

2.2 Lexical Distance

The standard Levenshtein distance is a string metric for measuring the amount of difference between two strings. This distance quantifies the number of changes (character based) to generate one text string (T) from the other (H). The algorithm works independently from the language that we are analyzing.

We used a Spanish Stemmer that stems words in Spanish based on a modified version of the Snowball algorithm[2].

Additionally, by using Levenshtein distance we defined a lexical distance and the procedure is the following:

- Each string T and H are divided in a list of tokens.
- The similarity between each pair of tokens in T and H is performed using the Levenshtein distance over stems.
- The string similarity between two lists of tokens is reduced to the problem of "bipartite graph matching", performed using the Hungarian algorithm (Kuhn, 1955) over this bipartite graph. Then, we found the assignment that maximizes the sum of ratings of each token. Note that each graph node is a token of the list.

The final score is calculated by:

$$finalscore = \frac{TotalSim}{Max(Length(T), Length(H))}$$

Where:
TotalSim is the sum of the similarities with the optimal assignment in the graph.
Length (T) is the number of tokens in T.
Length (H) is the number of tokens in H.

2.3 Wordnet Distance

Since, all datasets are in Spanish, we need to convert <T, H> pair to English. In the case of RTEs-Sp datasets, this action will backward to the English language (source).

Our ideal case would be to use EuroWordNet[3] to obtain the semantic information that we need, but we won't be able to access to this resource.

Thus, WordNet is used to calculate the semantic similarity between T and H. The following procedure is applied:

1. Word sense disambiguation using the Lesk algorithm (Lesk, 1986), based on Wordnet definitions.

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 by using two factors: length of the path and orientation of the path.

3. To obtain the final score, we use matching average.

The semantic similarity between two words is computed as:

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

Where: s,t are source and target words that we are comparing (s is in H and t is in T). Depth(s) is the shortest distance from the root node to the current node. LCS(s,t):is the least common subsume of s and t.

The matching average (step 3) between two sentences X and Y is calculated as follows:

$$MatchingAverage = 2 \times \frac{Match\ (X,Y)}{Length\ (X) + Length\ (Y)}$$

[2] http://snowball.tartarus.org/

[3] http://www.illc.uva.nl/EuroWordNet/

2.4 Longest Common Substring

Given two strings, T of length n and H of length m, the Longest Common Sub-string (LCS) problem (Dan, 1999) will find the longest string that is a substring of both T and H. It is found by dynamic programming.

$$lcs(T,H) = \frac{Length(MaxComSub(T,H))}{\min(Length(T), Length(H))}$$

3 Experimental Evaluation and Discussion of the Results

With the aim of exploring the differences among training sets and machine learning algorithms, we did many experiments looking for the best result to our system.

First, we converted the RTE4 and RTE5 datasets with Contradiction/Unknown/Entailment pair information to a binary True/False problem, named two-way problem.

Then, we used the following combination of datasets: RTE3-Sp, RTE4-Sp, RTE3-Sp+RTE4-Sp, SPARTE-Bal (balanced SPARTE Corpus with the same number of true and false cases), and SPARTE-Bal+ RTE3-Sp+RTE4-Sp. The training set SPARTE-Balanced was created by taking all true cases and randomly taking false cases, and then we build a balanced training set containing 1352 pairs, with 676 true and 676 false pairs.

We used four classifiers to learn every development set: (1) Support Vector Machine, (2) Ada Boost, (3) Multilayer Perceptron (MLP) and (4) Decision Tree using the open source WEKA Data Mining Software (Witten & Frank, 2005). In all the tables results we show only the accuracy of the best classifier.

The results obtained to predict RTE5-Sp in a two-way classification task are summarized in Table 1 below. In addition, table 2 shows our results reported in RTE two-way classification task by using with Cross Validation technique with 10 folds.

Dataset	*Classifier*	*Accuracy%*
RTE3-Sp+RTE4-Sp	SVM	60.83%
RTE3-Sp	SVM	60.50%
RTE4-Sp	MLP	60.50%
SPARTE-Bal+ RTE3-Sp+RTE4-Sp	MLP	60.17%
SPARTE-Bal	DT	50%
Baseline	-	50%

Table 1.Results obtained in two-way classification task.

Dataset	*Classifier*	*Accuracy%*
SPARTE-Bal	DT	68.19%
RTE3-Sp	SVM	66.50%
RTE3-Sp+RTE4-Sp	MLP	61.44%
RTE4-Sp	MLP	59.60%
SPARTE-Bal+ RTE3-Sp+RTE4-Sp	AdaBoost	56.83%
Baseline	-	50%

Table 2.Results obtained with Cross Validation 10 folds in two-way task.

The performance in all cases was clearly above those baselines. Only when using SPARTE-Bal we obtained a result equal to the baseline (50% true pairs and 50% false pairs).

The SPARTE-Balanced dataset yields the worst results, maybe because this dataset contains only pairs with QA task, and an additional reason, could be that SPARTE is syntactically simpler than PASCAL RTE. In that sense, some authors have reported low performance when using syntactically simpler datasets; for instance, by using BPI[4] dataset to predict RTEs datasets in English. Therefore, SPARTE seems to be not enough good training set to predict RTEs test sets.

The best performance of our system was achieved with SVM classifier with RTE3-Sp+RTE4-Sp dataset; it was 60.83% of accuracy. In the majority of the cases, SVM or MLP classifiers appear as ‘favorite’ in all classification tasks.

Surprisingly, in the two-way task, a slight and not statistical significant difference of 0.66% between the best and worst combination (except for SPARTE-Bal) of datasets and classifiers is found. So, it suggests that the combination of dataset and classifiers do not produce a strong impact predicting RTE5-Sp, at least, for these feature sets.

[4] http://www.cs.utexas.edu/users/pclark/bpi-test-suite/

Also, we observed that by including SPARTE-Bal to RTE3-Sp+RTE4-Sp dataset, the performance slightly decreases, although this difference was not statistical significant.

The results obtained in table 2(and table 4) with SPARTE-Bal and decision tree algorithm, are the best for cross-validation experiments. In fact, an accuracy of 68.19% was obtained, which is 18.19% bigger than the result obtained in table 1, and was statistical significant.

Finally, we assessed our system only over the SPARTE Corpus. First, we used cross validation technique with ten folds over SPARTE-Bal, testing over our four classifiers. Then, we tested SPARTE-Bal by splitting the corpus in training set (70%), and test set (30%).

The results are shown in the tables 4 and 5 below.

Classifier	*Accuracy%*
DT	68.19%
MLP	62.64%
AdaBoost	61.31%
SVM	60.35%
Baseline	50%

Table 4.Results obtained with Cross Validation 10 folds in two-way task to predict SPARTE.

Classifier	*Accuracy%*
DT	66.50%
AdaBoost	62.31%
SVM	59.60%
MLP	52.70%
Baseline	50%

Table 5.Results obtained with SPARTE with split 70%.

The results on cross-validation are better than those obtained on test set, which is most probably due to overfitting of classifiers.

Table 5 shows a good performance of 66.50%, predicting test set and using Decision trees. These results are opposed to the bad performance reported by SPARTE to predict RTEs datasets. Here, in fact, the syntactic complexity and original task do not change between train and test set; and it seems to be the main problem with the low performance of SPARTE in Table 1.

3.1 Related Work

Up to our knowledge, there are not available results of other teams that used SPARTE to predict RTE, or used RTEs applied to Spanish. However, some comparison with other results for Spanish could be done in AVE Challenge (Alberto Téllez-Valero et al., 2008; Ferrández et al., 2008; Castillo, 2008), but we will need to modify our system to test AVE 2008 test set and computing different metric for the ranking of the result.

On the other hand, comparing the results obtained with English in RTE5 TAC Challenge, we obtained a result not statistical significant with respect to the median score for English systems that is 61.17% of accuracy. Also, our system could be compared to independent-language RTE systems.

To finish, we think that several improvements could be done in order to improve the accuracy of the system, using syntactic features, more semantic information, and new external resources such as Acronyms database.

4 Opportunities for Collaboration

Our work is oriented to create a Textual Entailment System. Such system could be used by another system or teams of others Universities, as an internal module.

The entailment relations between texts or strings are very useful for a variety of Natural Language Processing applications, such as Question Answering, Information Extraction, Information Retrieval and Document Summarization.

For example, a RTE module could be used in a Question Answering system, where the answer of a question must be entailed by the text that supports the correctness of the answer; or an Automatic Summarization system could eliminate the passages whose meaning is already entailed by other passages and, by this way, reduce the size of the passages.

In addition, a question answering system could be enhanced by a RTE module, and also, these results are useful as Answer Validation System.

Our system was designed having in mind the interoperation among systems. Thus, the system inputs accept files in .xml format, and the output is text plain files and .xml files.

On the other hand, one of the resources that would allow this work advance is the EuroWordnet, because it could provide additional semantic information improving our semantic features, and so the performance of our system. Due to being an expensive and not freely available resource, we are avoiding using it, but we expect to be able to use it in the future. In section 3, we used Wordnet in order to obtain the relationship between two different concepts. Since Wordnet includes only synsets for English and not for Spanish, we have translated the <t,h> pairs to English using the online Microsoft Bing translator[5], in order to use Wordnet. As a result, a loss of performance was obtained. We believe that the use of EuroWordNet could benefit our semantic features.

Currently, we are keeping improving our system, and we are looking forward to get opportunities for collaboration with other teams of all the Americas.

5 Conclusion and Future work

In this paper we present an initial RTE System based for the Spanish language, based on machine learning techniques that uses some of the available textual entailment corpus and yields 60.83% of accuracy.

One issue found is that SPARTE Corpus seems to be not useful to predict RTEs-Sp datasets, because of the syntactic simplicity and the absence of task information different to QA task.

On the other hand, we found that a competitive result of 66.50%acc is reported by train and test set taken from SPARTE Corpus.

Future work is oriented to experiment with additional lexical and semantic similarities features and to test the improvements they may yield. Also, we must explore how to decrease the computational cost of the system. Our plan is keeping applying machine learning algorithms, testing with new features, and adding new source of knowledge.

References

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danillo Giampiccolo, and Bernardo Magnini. 2009. *The Fifth PASCAL Recognizing Textual Entailment Challenge.* In proceedings of Textual Analysis Conference (TAC). NIST, Maryland USA.

Adrian Iftene, Mihai-Alex Moruz.2009. *UAIC Participation at RTE5,* TAC 2009, Gaithersburg, Maryland, USA.

S. Mirkin, R. Bar-Haim, J. Berant, I. Dagan, E. Shnarch, A. Stern, and I. Szpektor.2009. *Bar-Ilan University's submission to RTE5,* TAC 2009, Gaithersburg, Maryland, USA.

Castillo, Julio. *Sagan in TAC2009: Using Support Vector Machines in Recognizing Textual Entailment and TE Search Pilot task.* TAC 2009, Gaithersburg, Maryland, USA.

Marie-Catherine de Marneffe, Bill MacCartney, Trond Grenager, Daniel Cer, Anna Rafferty and Christopher D. Manning. 2006. *Learning to distinguish valid textual entailments.* RTE2 Challenge, Italy.

F. Zanzotto, Marco Pennacchiotti and Alessandro Moschitti.2007. *Shallow Semantics in Fast Textual Entailment Rule Learners,* RTE3, Prague.

Ian H. Witten and Eibe Frank. 2005. "Data *Mining: Practical machine learning tools and techniques"*, 2nd Edition, Morgan Kaufmann, San Francisco, USA.

Anselmo Peñas, Alvaro Rodrigo, Felisa Verdejo. *SPARTE, a Test Suite for Recognising Textual Entailment in Spanish.* Cicling 2006, Mexico.

Peñas A., Rodrigo A., Sama V., and Verdejo F. *Overview of the Answer Validation Exercise 2006*, In-Working notes for the Cross Language Evaluation Forum Workshop (CLEF 2006), September 2006, Spain.

Ido Dagan, Oren Glickman and Bernardo Magnini. *The PASCAL Recognising Textual Entailment Challenge.* In Quiñonero-Candela, J.; Dagan, I.; Magnini, B.; d'Alché-Buc, F. (Eds.) Machine Learning Challenges. Lecture Notes in Computer Science , Vol. 3944, pp. 177-190, Springer, 2006.

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

Harold W. Kuhn, *The Hungarian Method for the assignment problem*, Naval Research Logistics Quarterly. 1955

Alberto Téllez-Valero, Antonio Juarez-Gonzalez, Manuel Montes-y-Gomez, Luis Villasenior-Pineda. *INAOE at QA@CLEF 2008:Evaluating Answer Validation in Spanish Question Answering*. CLEF 2008.

Julio J. Castillo. *The Contribution of FaMAF at QA@CLEF 2008.Answer ValidationExercise.*CLEF 2008.

Oscar Ferrández, Rafael Muñoz, and Manuel Palomar. *A Lexical Semantic Approach to AVE.* CLEF 2008.

[5] http://www.microsofttranslator.com/