SINAI at QA@CLEF 2007. Answer Validation Exercise

M.A. García-Cumbreras, J. M. Perea-Ortega
F. Martínez Santiago, L.A. Ureña-López
University of Jaén. Computers Department
SINAI Group
Campus Las Lagunillas, Ed. A3, E-23071, Jaén, Spain
{magc,jmperea,dofer,laurena}@ujaen.es

Abstract

This paper describes the first participation of the SINAI (Intelligent Systems of Access Information) group of the University of Jaén in the AVE task of QA@CLEF 2007. We have developed a system made up of training and classification processes, that uses machine learning methods (bbr, timbl). Based on lexical features it obtains good results, a 41% of QA accuracy.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms

Algorithms, Languages, Performance, Experimentation

Keywords

Question Answering, Answer Validation, Textual Entailment, Named Entity Recognition, QA@CLEF

1 Introduction

This document contains the description of the experiments carried out by SINAI group. We have developed an approach based on several lexical measures integrated by means of different machine learning models. More precisely, we have evaluated three features based on lexical similarity.

In order to calculate the semantic distance between two tokens (stems), we have tried several measures based on Lin's similarity measure. In spite of the relatively straightforward approach we have obtained a remarkable accuracy.

2 Approach description

Our system is based on a machine learning method that makes use of a binary classifier to solve the answer validation. In our approach we can distinguish two processes applied to this classifier: training and classification.

In the training process we have extracted several features for all the used training collections[1]. Previous results have been evaluated using the existing entailment judgements of these collections, and ML parameters have been adjusted.

We have trained the classifier obtaining a *learned model* that will be used later in the classification process.

In the classification process we also extract the same features used in the training process for each pair question-answer. The classification algorithm uses these features and the *learned model* obtained in the training process. This algorithm returns a boolean value (*correct* or *incorrect*) for each pair question-answer. Figure 1 describes the system architecture.

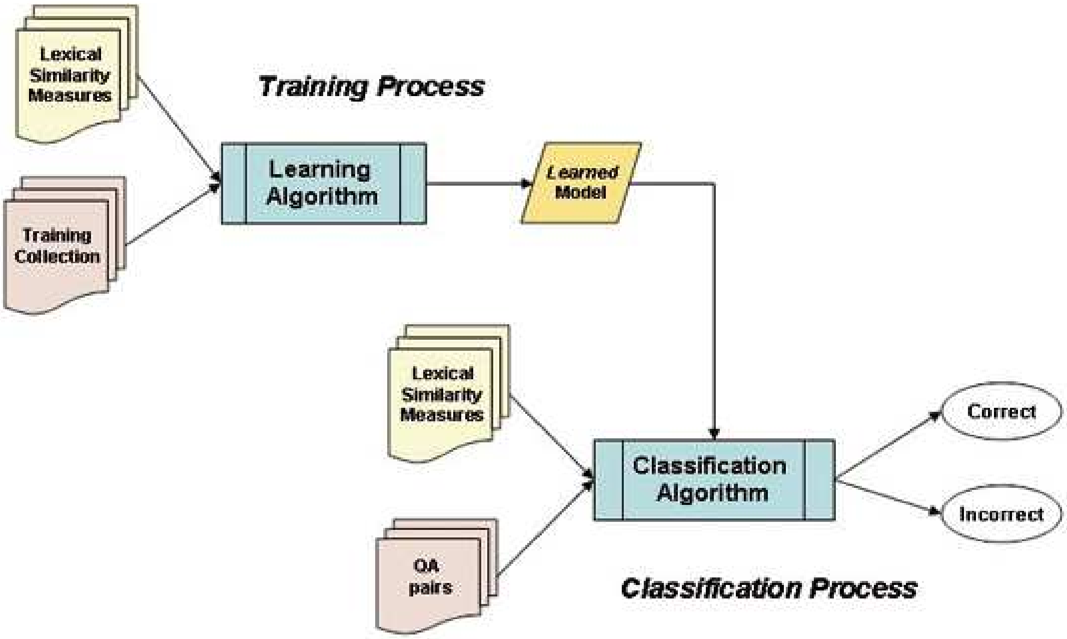

Figure 1: System architecture

The extracted features are related to the lexical similarity. In our experiments we have applied different lexical similarity measures. All these features are explained below.

2.1 Lexical similarity

This experiment approaches the textual entailment task, based on the extraction of a set of lexical measures, that check the existing similarity between the hypothesis-text pairs. Our approach is similar to [3] but the matching between pairs of words is relaxed by using the Lin's similarity measure[5] through Wordnet hierarchy. More concisely, we have applied simple matching, Binary Matching and Consecutive Subsequence Matching. In this task we have considered the answers as hypotheses and questions as texts.

Before the calculation of the different measures, the first step was to preprocess the pairs using the English *stopwords* list. Then, we have used the GATE[2] architecture to obtain the stems of tokens. Once the stems have been obtained, we have applied four different measures or techniques:

[1] Answer Validation Exercise training collection and Third Recognizing Textual Entailment Challenge (RTE3) training.

[2] http://gate.ac.uk/

- **Simple Matching**: this technique calculates the semantic distance between the stems of each question and its answer. If the distance exceeds a threshold, both stems are considered similar and the similarity weight value increases in one. The accumulated weight is normalized dividing it by the number of terms of the answer (hypothesis). In this experiment we have considered the threshold 0.5. The values of semantic distance measure range from 0 to 1. In order to calculate the semantic distance between two stems, we have tried several measures based on WordNet [1]. **Lin's similarity measure** [5] was shown to be best overall measures. It uses the notion of information content and the same elements as Jiang and Conrath's approach [4] but in a different fashion:

  $$sim_L(c_1, c_2) = \frac{2 \times \log p(lso(c_1, c_2))}{\log p(c_1) + \log p(c_2)}$$

  where c_1 and c_2 are synsets, *lso(*c_1*,*c_2*)* is the information content of their lowest superordinate (most specific common subsumer) and *p(c)* is the probability of encountering an instance of a synset c in some specific corpus [6]. The Simple Matching technique is defined in the following equation:

  $$SIM_{matching} = \frac{\sum_{i \in H} similarity(i)}{|H|}$$

  where H is the set that contains the elements of the answer (hypothesis) and *similarity(i)* is defined like:

  $$similarity(i) = \begin{cases} 1 & \text{if } \exists j \in T sim_L(i,j) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- **Binary Matching**: this measure is the same that the previous one, but modifying the *similarity* function:

  $$similarity(i) = \begin{cases} 1 & \text{if } \exists j \in T \; i = j \\ 0 & \text{otherwise} \end{cases}$$

- **Consecutive Subsequence Matching**: this technique relies on forming subsequences of consecutive stems in the answer (hypothesis) and matching them in the question (text). The minimal size of the consecutive subsequences is two, and the maximum is the maximum size of the answer. Every correct matching increases in one the final weight. The sum of the obtained weights of the matching between subsequences of a certain size or length is normalized by the number of sets of consecutive subsequences of the answer created for this length. These weights are accumulated and normalized by the size of the answer less one. The Consecutive Subsequence Matching technique is defined in the following equations:

  $$CSS_{matching} = \frac{\sum_{i=2}^{|H|} f(SH_i)}{|H| - 1}$$

  where SH_i is the set that contains the subsequences of the answer with i size or length and *f(*SH_i*)* is defined like:

  $$f(SH_i) = \frac{\sum_{j \in SH_i} matching(j)}{|H| - i + 1}$$

  where

  $$matching(i) = \begin{cases} 1 & \text{if } \exists k \in ST_i \; k = j \\ 0 & \text{otherwise} \end{cases}$$

  where ST_i represents the set that contains the subsequences with i size from question (text).

- **Trigrams**: this technique relies on forming trigrams of words in the answer and matching them in the question. If an answer trigram matches in question, then the similarity weight value increases in one. The accumulated weight is normalized dividing it by the number of trigrams of the answer.

Table 1: Results with TiMBL and BBR classifiers.

Experiment	Classifier	Train Data	F measure	qa accuracy
Exp1	BBR	RTE-3	0.19	0.08
Exp2	TiMBL	RTE-3 and AVE-2007	0.37	0.41

3 Experiments and Results

The algorithms used in the experiments as binary classifiers are two: *Bayesian Logistic Regression* (BBR)[3] and TiMBL [2]. Both algorithms have been trained with the *devel* data provided by the organization of the Pascal challenge (RTE-3) and the AVE task of CLEF.

As it has been explained in previous sections, a model is generated via the supervised learning process. This model is used by the classification algorithm, which will decide whether an answer is entailed by the given snippet or not.

Table 1 shows two official results:

where:

- **Exp1** uses three features: three lexical similarities ($SIM_{matching}$ + $CSS_{matching}$ + Trigrams). The model has been trained using the devel data provided by the organization of the Pascal challenge, RTE-3, and the ML method used was BBR. comparison.
- **Exp2** uses the same three features. The model has been trained using the devel data provided by the organization of the Answer Validation Exercise task, AVE-2007, and the devel data provided by the organization of the Pascal challenge, RTE-3. The ML method used was TiMBL.

As we expected, the best result is obtained by means of the use of both devel collections, RTE-3 and AVE-2007, and the use of TiMBL. We have to investigate why both results are too different.

4 Conclusions and Future work

In spite of the simplicity of the approach, we have obtained remarkable results: each set of features has reported relevant information, concerning to the entailment judgement determination. Our experiments approach the textual entailment task being based on the extraction of a set of lexical measures that show the existing similarity between the hypothesis-text pairs.

We have applied simple matching, Binary Matching and Consecutive Subsequence Matching, but the matching between pairs of words is relaxed by using the Lin's similarity measure through Wordnet hierarchy.

Finally, we want to implement a hierarchical architecture based on constraint satisfaction networks. The constraints will be given by the set of available features and the maintenance of the integrity across the semantic interpretation process.

5 Acknowledgments

This work has been partially supported by a grant from the Spanish Government, project TIMOM (TIN2006-15265-C06-03), and the RFC/PP2006/Id_514 granted by the University of Jaén.

References

[1] Alexander Budanitsky and Graeme Hirst. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. 2001.

[3] http://www.stat.rutgers.edu/˜madigan/BBR

[2] Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. Timbl: Tilburg memory based learner, version 1.0, reference guide., 1998.

[3] Oscar Ferrandez, Daniel Micolo, Rafael Mu noz, and Manuel Palomar. Técnicas léxico-sintácticas para reconocimiento de inmplicación textual. . *Tecnologías de la Informacón Multilingüe y Multimodal. In press.*, 2007.

[4] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, Taiwan, 1997.

[5] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, 1998.

[6] Philip Resnik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montreal, 1995.