The Effect of Entity Recognition on Answer Validation

Álvaro Rodrigo, Anselmo Peñas, Jesús Herrera, Felisa Verdejo

Departamento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia
Madrid, Spain
{alvarory, anselmo, jesus.herrera, felisa}@lsi.uned.es

Abstract. The Answer Validation Exercise (AVE) 2006 is aimed at evaluating systems able to decide whether the responses of a Question Answering (QA) system are correct or not. Since most of the questions and answers contain entities, the use of a textual entailment relation between entities is studied here for the task of Answer Validation. We present some experiments concluding that the entity entailment relation is a feature that improves a SVM based classifier close to the best result in AVE 2006.

1 Introduction

The Answer Validation Exercise (AVE) [8] of the Cross Language Evaluation Forum (CLEF) 2006 is aimed at developing systems able to decide whether the responses of a Question Answering (QA) system are correct or not using a Textual Entailment approach [3]. The AVE 2006 test corpus contains hypothesis-text pairs where the hypotheses are built from the questions and answers of the real QA systems, and the texts are the snippets given by the systems to support their answers. Participant systems in AVE 2006 must return a value YES or NO for each hypothesis-text pair to indicate if the text entails the hypothesis or not (i.e. the answer is correct according to the text).

The questions and answers of the QA Track at CLEF contain many entities (person names, locations, numbers, dates...) due to the nature of the questions in this evaluation: 75% of the questions in Spanish were factoids in the previous campaign [10]. For this reason, we studied the consideration of an entity entailment relation in combination with a SVM classifier to solve the Answer Validation problem.

Section 2 describes the process of entity detection and entailment decision. Section 3 describes the experimental setting combining the entity based entailment decision with a baseline SVM system. Section 4 shows the results of the experiments together with an error analysis. Finally, we give some conclusions and future work in Section 5.

2 Entailment between entities

The first step for considering an entailment relation between entities is to detect them in a robust way that minimizes the effect of errors in the annotation.

The second step is the definition and implementation of an entailment relation between entities (numeric expressions, temporal expressions and named entities in our case).

Next subchapters describe in detail these steps involved in the decision of entity entailment.

2.1 Entity recognition

We have used the FreeLing[1] [2] Name Entity Recognizer (NER) to tag numeric expressions (NUMEX), named entities (NE) and time expressions (TIMEX) of both text and hypothesis. The first approach was to use the categories given by the tool. However, the ambiguity of these expressions was not always solved correctly by the tool. Figure 1 shows an example of this problem. The expression 1990 is a year but it is recognized as a numeric expression in the hypothesis. However, the same expression is recognized as a temporal expression in the text and, therefore, the expression in the hypothesis cannot be entailed by it. This kind of errors that are not consistent between texts and hypotheses, breaks the possible entailment relation between entities.

This fact led us to ignore the entity categorization given by the tool and assume that text and hypothesis are related and close texts where same expressions must receive same categories, without the need of disambiguation. Thus, all detected entities received the same tag.

<**t**>...Irak invadió Kuwait en <**TIMEX**>agosto_de_1990<**/TIMEX**>...<**/t**>
<**h**>Irak invadió Kuwait en <**NUMEX**>1990<**/NUMEX**><**/h**>

Fig. 1. Example of error disambiguating the entity type

2.2 Entailment relations between entities

Once the entities of the hypothesis and the text are detected, the next step is to determine the entailment relations between the entities in the text and the entities in the hypothesis.

We defined the following entailment relations between entities.

1. A Named Entity E_1 entails a Named Entity E_2 if the text string of E_1 contains the text string of E_2. For example, *Yaser_Arafat* entails *Arafat*, but *Arafat* does not entail *Yaser_Arafat* (see example in Figure 2). Sometimes some characters change in different expressions of the same entity as, for example, in a proper noun with different wordings (e.g. *Yasser, Yaser, Yasir*). To detect the entailment in these situations, when the previous process fails, we implemented the edit distance of Levenshtein [7] following [6]. Thus, if two Named Entities differ in less than 20%, then we assume that there exists an entailment relation between these entities.

[1] http://garraf.epsevg.upc.es/freeling/

2. A Time Expression T_1 entails a Time Expression T_2 if the time range of T_1 is included in the time range of T_2. For example, *26 April 1986* entails *April 1986* but not in the other sense. The first approach to implement this is to consider time expressions as strings and check if one string contains the other in a similar way to the Named Entities. However, there are several ways to express the same date with different text strings. The second approach is to consider the normalized form of the time expressions. However, the normalization depends on the correct detection of the entity type. Thus the best approach is to give the two chances for the entailment, the original string and a normalized form. Figure 2 shows a positive example of entailment between temporal expressions.
3. A numeric expression N_1 entails a numeric expression N_2 if the range associated to N_2 encloses the range of N_1. In [5] we considered the units of the numerical expressions where the unit of N_1 must entail the unit of N_2. For example, *17,000,000 citizen* entails *more than 15 million people*. However, sometimes the detection of units needs some kind of anaphora resolution and we ignored the units for the experiments described here. Figure 2 shows an example of numeric expression where the normalization is considered allowing the detection of the entailment relation.

<**t**>...presidida por <**ENTITY**>Yaser_Arafat<**/ENTITY**>...<**/t**>
<**h**><**ENTITY**>Arafat<**/ENTITY**> preside la ...<**/h**>

<**t**>...durante la fuga de <**ENTITY**>Chernobyl<**/ENTITY**> el
<**ENTITY**>26_de_abril_de_1986<**/ENTITY**>...<**/t**>
<**h**>La catástrofe de <**ENTITY**>Chernobyl<**/ENTITY**> ocurrió en
<**ENTITY**>abril_de_1986<**/ENTITY**><**/h**>

<**t**>Los seis países citados forman parte
de los <**ENTITY**>diez<**/ENTITY**> europeos que ingresaron ya... <**/t**>
<**h**>...está formada por <**ENTITY**>10<**/ENTITY**> países.<**/h**>

Fig. 2. Pairs that justify the process of entailment.

3 Experimental setting

The system developed for the AVE 2006 in Spanish is based on the ones developed for the First [3] and the Second [1] Recognizing Textual Entailment (RTE) Challenges for English.

In the system here described, the basic ideas from the ones presented to the RTE Challenges were kept, but the new system was designed and developed according to the available resources for the Spanish language, lacking some subsystems implemented in English such, for example, dependency analysis.

The proposed system is based on lemmatization. The system accepts pairs of text snippets (text and hypothesis) at the input and gives a boolean value at the output:

YES if the text entails the hypothesis and NO otherwise. This value is obtained by the application of a learned model by a SVM classifier. System components, are the following:

– *Linguistic processing*: The lemmas of every text and hypothesis pairs are obtained using Freeling.
– *Sentence level matching*: A plain text matching module calculates the percentage of words, unigrams (lemmas), bigrams (lemmas) and trigrams (lemmas), respectively, from the hypothesis entailed by lexical units (words or n-grams) from the text, considering them as bags of lexical units.
– *Entailment Decision*: A SVM classifier, from Yet Another Learning Environment (Yale 3.0) [4], was applied in order to train a model from the development corpus given by the organization and to apply it to the test corpus.

3.1 SVM with baseline attributes

The SVM model was trained by means of a set of features obtained from the sentence level matching module for every pair <text, hypothesis>:

1. Percentage of words of the hypothesis present in the text (treated as bags of words).
2. Percentage of unigrams (lemmas) of the hypothesis present in the text (treated as bags of unigrams).
3. Percentage of bigrams (lemmas) of the hypothesis present in the text (treated as bags of bigrams).
4. Percentage of trigrams (lemmas) of the hypothesis present in the text (treated as bags of trigrams).

The first experiment evaluated the results obtained by this baseline system.

3.2 Entailment decision based only on entities entailment

The thesis we follow in the recognition of textual entailment is that all the elements in the hypothesis must be entailed by elements of the supporting text. In special, all the entities in the hypothesis must be entailed by entities in the supporting text. Therefore, the system assumes that if there is an entity in the hypothesis not entailed by one or more entities in the text, then the answer is not supported and the system must return the value NO for that pair.

However, in pairs where all the entities in the hypothesis are entailed, there is not enough evidence to decide if the value of the pair is YES or NO.

In this experiment we decided to evaluate the results obtained when the default value is always YES except if there exist entities not entailed in the hypothesis (evidence of a not supported answer).

3.3 Check entities entailment before SVM classification

As we argued above (section 3.2), the information about entities is useful to detect some pairs without entailment. However, there is no enough evidence to decide the entailment value in the pairs where all the entities in the hypothesis are entailed. Therefore, a solution is to apply the SVM setting (section 3.1) only in this case. First, the system assigns the value NO to the pairs with entities not entailed in the hypothesis and second, the rest of pairs are classified with the baseline SVM system.

3.4 SVM classification adding the attribute of entity entailment

The last experiment was to use the information of entailment between entities as an additional attribute in the SVM classifier, in a similar way we already tested in English with numeric expressions in the system of the RTE2 Challenge [5].

4 Results and error analysis

The proposed system has been tested over the Spanish test set of the AVE 2006. Table 1 shows the precision, recall and F-measure for the different settings compared with a baseline system that always returns YES and the system that obtained the best results at AVE 2006 (COGEX) [9].

Table 1. Results of the experiments compared with the best and the baseline systems.

System	F-measure	Precision over YES	Recall over YES
Best AVE 2006 System	0.61	0.53	0.71
SVM classification adding the attribute of entity entailment	0.60	0.49	0.77
Entailment decision based only on entities entailment	0.59	0.46	0.83
SVM with baseline attributes	0.56	0.47	0.71
Check entities entailment before SVM classification	0.55	0.46	0.71
100% YES Baseline	0.45	0.29	1

The system based on SVM with the basic attributes obtained an F of 0.56 which is not bad compared with the baseline (F of 0.45). However, this result is worse than considering only the entailment relation between entities (F of 0.59). This is due to the higher recall obtained (0.83) after giving a YES value to the pairs that passed the entity entailment test. As it was mentioned before, the answer validation decision based only on the entailment between entities has a good performance for detecting the wrong answers (achieving an 89% of precision in the detection of pairs without entailment), but it does not provide enough evidence for most of the pairs. For this reason we used the system based on SVM with the basic attributes to decide on the rest of pairs. However, this setting obtained worse results (F of 0.55). The appropriate configuration was to include the entailment relation between entities as an additional attribute inside the SVM, obtaining an F of 0.60, close to the 0.61 obtained by COGEX.

Most of the errors in the entailment relation between entities were due to a wrong detection of the entities. In many cases the QA systems changed the original text, ignoring the writing of the named entities and returning all the words of the supporting snippets in lower case. In these cases, the NER tool cannot recognize any named entity in the text and the entities in the hypothesis are not entailed.

In some other cases the answers are given in capital letters and the NER tool recognize all words as named entities despite of they are not usually named entities. Therefore there are some false entities in the hypothesis that cannot be entailed by the text.

To solve these problems we need more robust NER tools and, at same time, the QA systems must take into account that their output can be taken as input by answer validation systems. QA systems should return their responses as they appear in the original texts.

5 Conclusions and future work

We have defined an entailment relation between entities and we have tested its use in an Answer Validation system. The addition of this relation as an additional attribute in a SVM setting improves the results of the system close to the best results in the AVE 2006.

Compared with the best in AVE 2006, our system gets higher recall and lower precision suggesting that we still have room to work on more restrictive filters to detect pairs without entailment.

Future work is oriented to the development of more robust NER tools and richer implementations of the entailment relations between entities.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Technology within the project R2D2-SyEMBRA. (TIC-2003-07158-C04-02), the Regional Government of Madrid under the Research Network MAVIR (S-0505/TIC-0267) and a PhD grant by UNED.

References

1. R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and Idan Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venezia, Italy*, April 2006.

2. X. Carreras, I. Chao, L. Padró, and M. Padró. FreeLing: An Open-Source Suite of Language Analyzers. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04). Lisbon, Portugal*, 2004.
3. I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment, Southampton, UK*, pages 1–8, April 2005.
4. S. Fischer, R. Klinkenberg, I. Mierswa, and O. Ritthoff. Yale 3.0, Yet Another Learning Environment. User Guide, Operator Reference, Developer Tutorial. Technical report, University of Dortmund, Department of Computer Science. Dortmund, Germany., 2005.
5. J. Herrera, A. Peñas, Á. Rodrigo, and F. Verdejo. UNED at PASCAL RTE-2 Challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venezia, Italy*, pages 38–43, April 2006.
6. Jesús Herrera, Anselmo Peñas, and Felisa Verdejo. Textual entailment recognition based on dependency analysis and *WordNet*. In Joaquin Quiñonero Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 231–239. Springer, 2005.
7. V. I. Levensthein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet Physics - Doklady*, volume 10, pages 707–710, 1966.
8. Anselmo Peñas, Álvaro Rodrigo, Valentín Sama, and Felisa Verdejo. Overview of the answer validation exercise 2006. In *Lecture Notes in Computer Science, Springer-Verlag*, 2007.
9. Marta Tatu, Brandon Iles, and Dan Moldovan. Automatic Answer Validation using COGEX. In *Working Notes. CLEF 2006 Workshop, 20-22 September, Alicante, Spain*, 2006.
10. A. Vallin, B. Magnini, D. Giampiccolo, L. Aunimo, C. Ayache, P. Osenova, A. Peñas, M. de Rijke, B. Sacaleanu, D. Santos, and R. Sutcliffe. Overview of the CLEF 2005 Multilingual Question Answering Track. In C. Peters, F.C. Gey, J. Gonzalo, H. Müller, G.J.F. Jones, M. Kluck, B. Magnini, and M. De Rijke, editors, *Accessing Multilingual Information Repositories*, volume 4022 of *Lecture Notes in Computer Science*, pages 307–331. Springer, September 2006.