Applying Dependency Trees and Term Density for Answer Selection Reinforcement

Manuel Pérez-Coutiño[1], Manuel Montes-y-Gómez[2], Aurelio López-López[2],
Luis Villaseñor-Pineda[2] and Aarón Pancardo-Rodríguez[1]

[1]Vanguard Engineering Puebla (VEng)
Recta a Cholula 308-4, CP 72810, San Andrés Cholula, Pue., México
[1]{mapco, apancardo}@v-eng.com
[2]Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)
Luis Enrique Erro No. 1, CP 72840, Sta. Ma. Tonantzintla, Pue., México.
[2]{mmontesg, allopez, villasen}@inaoep.mx

Abstract. This paper describes the experiments performed for the QA@CLEF-2006 within the joint participation of the eLing Division at VEng and the Language Technologies Laboratory at INAOE. The aim of these experiments was to observe and quantify the improvements in the final step of the Question Answering prototype when some syntactic features were included into the decision process. In order to reach this goal, a shallow approach to answer ranking based on the term density measure has been integrated into the weighting schema. This approach has shown an interesting improvement against the same prototype without this module. The paper discusses the results achieved, the conclusions and further directions within this research.

1 Introduction

Over the last years, Question Answering (QA) system research has shown an incremental growing both in interest as well as in complexity. Particularly, QA for Spanish has been formally evaluated within the CLEF initiative during the last three years, increasing the number of participant groups and, consequently, the proposed systems. Those QA systems have explored different approaches to cope with the QA problem, analyzing several methodologies from purely data-driven [7, 11] to in-depth natural language processing (NLP) [5]. Despite the methods used in these approaches, the improvement of factoid question resolution has been measured.

This paper presents the prototype developed as a shared effort between the recently formed eLing Division at VEng and the Language Technologies laboratory at INAOE. This approach continues with the previous work of the authors [9] to cope with factoid questions resolution. The aim of these experiments was to observe and quantify the possible improvement at the answer selection step of a QA prototype, as a consequence of introducing some syntactic features to the decision process. In order to reach this goal, the following key points have been included in the QA prototype: i) a syntactic parser based on dependency grammars; ii) a shallow technique to weigh the number of question terms which have a syntactic dependency to one candidate answer within a relevant passage to the given question (*term density*); iii) a formula for merging the term density of each candidate answer and the weights gathered in the previous steps to obtain its total weight. Finally, the answer selection process arranges candidate answers based on their weights, selecting the top-*n* as the QA system answers. The approach described in this document has shown an interesting improvement up to 15% against the same QA prototype without this module.

The rest of the paper is organized as follows: Section 2 summarizes the architecture of the prototype; section 3 exposes the details of the new elements within the prototype architecture; section 4 discusses the results achieved by the system. Finally, section 5 draws conclusions and discusses further work.

2 Prototype Architecture

As stated before, the system developed for QA@CLEF 2006 is based on the previous work of the authors [9], where the most important modification relies on the inclusion of syntactic features to the decision process at the answer selection module. Figure 1 shows the main blocks of the system. It could be noticed that factoid and definition questions are handled independently. This report is focused on the factoid questions resolution process, while details of the processes involved in the creation and use of definition patterns aimed to answering definition questions can be found in [4].

Factoid question treatment relies on a typical QA architecture, including the following steps: *question processing*, *documents processing*, *searching*, and finally *answer extraction and selection*. Next paragraphs summarize the initial steps of the prototype whilst section three and four discuss the new ones.

2.1 Question Processing

Our prototype implements this step following a straight forward approach, parsing the question with a set of heuristic rules in order to get its semantic class. The MACO POS tagger [3] is used to tag the question, as well as to identify and classify its named entities. This information will be used later, during the searching step, mainly to match questions and candidate answer context, contributing to the weight schema.

2.2 Documents Processing

The processing of target documents first applies MACO [3] to the whole document collection (similar to the question processing). In the second part of the process, and in parallel to the first one, the whole document collection is tagged with the FDG Parser from Conexor, which is based on the *Functional Dependency Grammar* discussed in [12]. The final part of this step is performed by the JIRS [6] passage retrieval system (PRS), which create the index for the searching process. The index gathered by JIRS and the tagged collection are aligned phrase by phrase for each document in the collection. This way, the system could retrieve later the relevant passages for a given question with JIRS, and then use their tagged form for the answer extraction process.

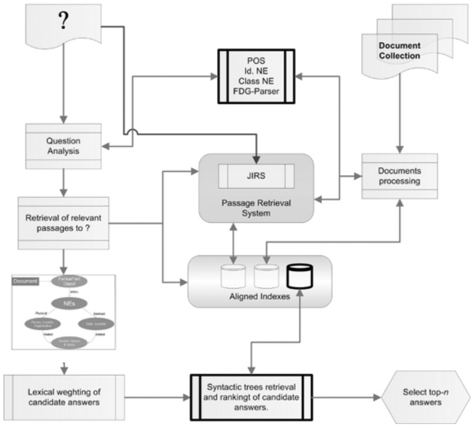


Fig. 1. Block diagram of the system. Factoid and definition questions are treated independently.

2.3 Searching

This step is performed in two parts. The first one consists in retrieving the relevant passages for the given question. This step is performed by JIRS, taking as input the question without previous processing. JIRS ranks the retrieved passages based on the computation of a weight for each passage. This weight is related to the larger n-gram structure of the question that can be found in the passage itself. The larger the n-gram structure, the greater the weight of the passage. Details of JIRS metrics can be found in [6]. The second part of the process requires the POS and Parsing tagged forms of each passage in order to gather the representation used to extract candidates answers. Tagged passages are represented as described in [8] where each retrieved passage is modeled by the system as a factual text object whose content refers to several named entities even when it is focused on a central topic. The model assumes that the named entities are strongly related to their lexical context, especially to nouns (subjects) and verbs (actions). Thus, a passage can be seen as a set of entities and their lexical context. Such representation is used later in order to match question's representation with the best set of candidates gathered from passages.

2.4 Answer Extraction

One of the drawbacks found in our previous work was the lost of precision during answer extraction. Once the system gathers a set of candidate answers, it computes for each candidate answer the lexical weight [9] (see formula 1) in order to rank the best one and then selects the top-*n* as the system answers. However, there were situations where several candidate answers could have the same weight. In order to minimize such situations syntactic information has been included, computing an additional weight which is later combined with the lexical weight to get the final rank of the candidate answers. This concept is explained in detail in the next section.

$$\omega_{lex}(A) = \frac{t_q}{n} * \left(\frac{|NE_q \cap NE_A|}{|NE_q|} + \frac{|C_q \cap C_A|}{|C_q|} + \frac{F_A(P_i)}{F_A(P)} + \left(1 - \frac{P_i}{k-1}\right) \right) \tag{1}$$

i=1..k; k=number of passages retrieved by JIRS

3 Adding Syntactic Features

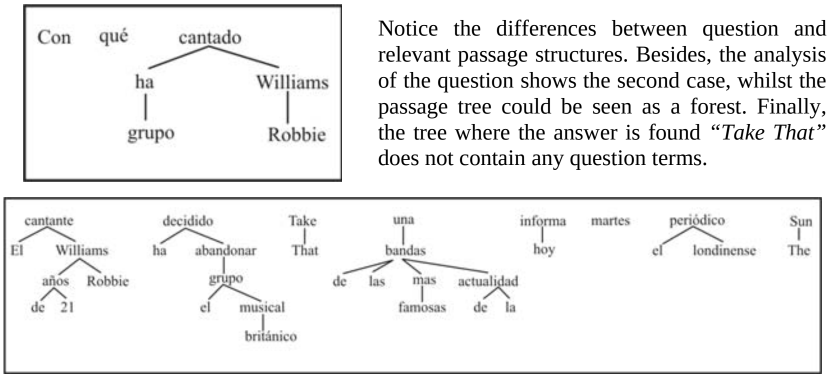
In order to improve the precision at the answer selection step, we have experimented with the inclusion of a shallow approach based on the use of some syntactic features. The main characteristic of this approach relies on the kind of information used to compute an additional weight to rearrange the set of candidate answers previously selected by a lexical supported method. This approach is flexible given that one of the factors taken in mind was to realize the amount of errors that a parser could yield.

3.1 Key Observations of the Method

There are different approaches to the use of syntactic information within the QA task. Some of them try to found a direct similarity between question structure and those of the candidate answers. This kind of similarity is commonly supported by a set of syntactic patterns which are expected to be matched by the answers. Some systems applying this approach are described in [1, 2, 10]. Other proposals, like the one presented by Tanev et al. [11] apply transformations to the syntactic tree of the question in order to approximate it to the trees of the relevant passages, where it is supposed that an answer is present.

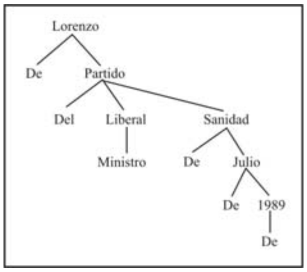
Despite the degree of effectiveness of those approaches, their principal drawback comes up when the parsing is deficient. With this limitation in mind, and trying to work around the gap of parsers for Spanish, this proposal is supported by the following observations of the analysis obtained through DFG parser (see figure 2).

1. There are important structural differences between dependency trees of a given question and those gathered from their relevant passages.
2. For a given question, its dependency tree states both, functional and structural dependencies starting from the main verb within the sentence, clearly delimiting other relations like subject, agent, object, etc. Contrasting this fact, dependency trees gathered from the relevant passage of the given question, could be seen as a forest, where functional and structural relations –which could possible lead the process to the accurate answer– are broken from one tree to another.
3. Finally, dependency trees gathered from the relevant passages to a given question could enclose a high number of question terms related to several candidate answers.



Notice the differences between question and relevant passage structures. Besides, the analysis of the question shows the second case, whilst the passage tree could be seen as a forest. Finally, the tree where the answer is found *"Take That"* does not contain any question terms.

Fig. 2a. Example of syntactic trees gathered from a question and one of its relevant passages.



This example shows the tree gathered from the relevant passage to the question: *¿Qué político liberal fue ministro de Sanidad italiano entre 1989 y 1993?*

Notice that the accurate answer *"De Lorenzo"*, occurs within this tree, and relates several question terms to it.

Fig 2b. Example of one tree with several question terms related to its candidate answer (De Lorenzo).

3.2 Term Density

In order to cope with these observations, we propose a straight forward metric to capture and weigh up the question terms which are nearest to a candidate answer within a relevant passage. Formula 2 shows those relations, which we have named the *term density* within a dependency tree.

The algorithm applied in order to compute the term density for a candidate answer involves the following steps.

```
1. For each relevant passage of a given question
1.1. Retrieve the dependency tree for that passage
1.2. For each candidate answer within the passage
1.2.1. Retrieve the sub tree where the candidate answer occurs
1.2.2. Apply formula 2 ($\delta_q$)
1.2.3. Compute the maximum $\delta_q$ for each candidate answer, then preserve it if it is greater than 0.5, in other case, $\delta_q$ =0
```

Given:
$Q=\{t_1,t_2...t_n\}$, the questions terms (lemmas)
$W=\{w_1,w_2...w_k\}$, the passages' terms (lemmas)
$C_i \subset W$, the terms (lemmas) of the *i-th* candidate answer.

A sub tree $x_\alpha(w_1,...,*,...,w_h): w_1...w_h$ depends on x_α (2)

Then, $\delta_q(C_i)=\frac{1}{n}\sum_{\forall t\in Q} f(t_j) \Leftrightarrow \forall w\in C_i, \langle w, x_\alpha\rangle$;

$$f(t_j)=\begin{cases}1, \langle t_j, x_\alpha\rangle \text{ ó } t_j(*)\\ 0, \mathit{else}\end{cases}$$

Finally, computes the total weight of each candidate answer by formula 3.

$$\omega(C_i)=\alpha\omega_{lex}(C_i)+\beta\delta_q(C_i) \quad (3)$$

Where ω_{lex} is the result of computing formula 1; α and β coefficients has been selected experimentally, giving them values $\alpha=1/3$, and $\beta=2/3$. This way the syntactic weight has a greater confidence.

4 Experiments and Results

Through experiments performed over several training sets, including the evaluation set of last year QA@CLEF, it could be observed a significant improvement in the final answer selection step. For the case of training with the QA@CLEF-2005 evaluation set, the accuracy of the system was increased over 7% for factoid questions, whilst the improvement in temporal restricted factoid questions achieved a 15%. These percentages represent a good progress giving that over the last years the rate of improvements in the results of Spanish factoid questions evaluation has been gradually. Table 1 shows some examples of the increasing rank for candidate answers with high term density.

Table 1. Examples of rank increasing for candidates' answers with high term density.

Question No.	Candidate and Right Answer	Previous Rank TO New Rank	$w_{lex}(C_i)$	$\delta_q(C_i)$	$w(C_i)$
29	De Lorenzo	5th TO 1st	0.5472	0.5000	0.5157
115	64 (días)	15th TO 1st	0.5440	0.5714	0.5622
139	Yoweri Kaguta Museveni	10th TO 1st	0.9367	0.8888	0.9048
161	Jacques Delors	3rd TO 1st	0.8505	0.8000	0.8168

The evaluation of QA systems at CLEF 2006 campaign included factoid, definition, temporal and list questions. The organizers provide to participants with a set of 200 unclassified questions, i.e. there were not markers to indicate the type of expected answer. Another novelty was that teams must provide answers with the specific passage where the answer was extracted from. The later was used to facilitate the evaluation of answers.

We participate in the evaluation with one run. The configuration applied considers three classes of possible answers, Date, Quantity and Proper Nouns; the system analyzes the first 100 1-line passages retrieved by JIRS; the lexical context used was formed with nouns, named entities, verbs, and adjectives, and the size of the window context is 8 words. Table 2 shows the results of the evaluation.

Despite the fact that our results (for factual questions) were only over 2% better than last year, we believe that the approach described could be a good starting point to the introduction of syntactic information to the answer selection process. Some errors observed while training include the confusion of accurate answers by candidate answers that have a term density similar or greater to the right answer. Next direction in our research could include the use of structural relations to get a better discrimination of candidate answers.

Table 2. Results of evaluation at QA@CLEF2006

Run	***Vein061eses***
Right	80 (45F + 35D + 0 TRF)
Wrong	102
ineXact	3
Unsupported	5
Overall Accuracy	42.11%
Factoid Questions	30.82%
Definition Questions	83.33%
Temporally Restricted Factoid Questions	0%
Overall Confidence Weighted Score (CWS) over F, D, and TR	0.33582

5 Conclusions

This work has presented a method for the inclusion of syntactic information within the final process of answer selection in a QA system. The approach relies in the use of a flexible metric which allows measuring the amount of question terms which have a syntactic dependence to the candidate answers. Although official evaluation does not reach the expectation, preliminary results have demonstrated a significant improvement in the answer selection step. This leads us to think that it could be possible to apply syntactic information in several ways in order to cope with the problem of partial or even more, deficient syntactic trees (in particular dependence trees).

Despite the low increasing in the official evaluation, the methods applied at different steps of the QA process are stable; this conclusion can be inferred from the fact that the prototype has reached its last year performance.

It is important to realize that the additions to our QA prototype presented in this document are limited by the previous processes. This means that the proposed method is not able to extract new candidate answers. Therefore our next steps into QA systems development must be done in the direction of improving system's recall.

Acknowledgements. This work was done under partial support of Vanguard Engineering Puebla SA de CV, CONACYT (Project Grants U39957-Y and 43990), SNI-Mexico, and the Human Language Technologies Laboratory at INAOE. We also like to thanks to the CLEF as well as EFE agency for the resources provided.

References

1. Aunimo L., Kuuskoski R.: Question Answering Using Semantic Annotation. In Peters C. (ed.): 6th Workshop of the Cross-Language Evaluation Forum, Vienna, Austria (2005).
2. Bertagna F., Chiran L., Simi M.: QA at ILC-UniPi: Description of the Prototype. In Peters C. and Borri F. (eds.): Working Notes for the Cross Language Evaluation Forum Workshop (CLEF-2004), Bath, England (2004).
3. Carreras, X. and Padró, L.: A Flexible Distributed Architecture for Natural Language Analyzers. In Proceedings for the LREC'02, Las Palmas de Gran Canaria, Spain (2002).
4. Denicia-Carral C., Montes-y-Gómez M., Villaseñor-Pineda L. and García-Hernández R.: A Text Mining Approach for Definition Question Answering. In Proceedings for the Fifth International Conference on Natural Language Processing, Turku, Finland (2006).
5. Ferrés D., Kanaan S., González E., Ageno A., Rodríguez H.: The TALP-QA system for Spanish at CLEF 2005. In Peters C. (ed.): 6th Workshop of the Cross-Language Evaluation Forum, Vienna, Austria (2005).
6. Gómez-Soriano, J.M., Montes-y-Gómez, M., Sanchis-Arnal, E., Rosso, P.: A Passage Retrieval System for Multilingual Question Answering. In 8th International Conference on Text, Speech and Dialog, TSD, Springer LNAI (2005).
7. Montes-y-Gómez, M., Villaseñor-Pineda L., Pérez-Coutiño, M., Gómez-Soriano J.M., Sanchis-Arnal, E., and Rosso P.: INAOE-UPV Joint Participation at CLEF 2005: Experiments in Monolingual Question Answering. In Peters C. (ed.): 6th Workshop of the Cross-Language Evaluation Forum, Vienna, Austria (2005).
8. Pérez-Coutiño M., Solorio T., Montes-y-Gómez M., López-López A. and Villaseñor-Pineda L.: Toward a Document Model for Question Answering Systems. In Jesus Favela et al. (eds.): Advances in Web Intelligence: Second International Atlantic Web Intelligent Conference AWIC04, LNAI 3034, pp. 145-154, Springer-Verlag (2004).
9. Pérez-Coutiño M., Montes-y-Gómez M., López-López A. and Villaseñor-Pineda L., Experiments for tuning the values of lexical features in Question Answering for Spanish. In Peters C. (ed.): 6th Workshop of the Cross-Language Evaluation Forum, Vienna, Austria (2005).
10. Roger S., Ferrández S., Ferrández A., Peral J., Llopis F., Aguilar A., and Tomás D.: AliQAn, Spanish QA System at CLEF-2005. In Peters C. (ed.): 6th Workshop of the Cross-Language Evaluation Forum, Vienna, Austria (2005).
11. Tanev H., Kouylekov M., Magnini B., Negri M., and Simov K.: Exploiting Linguistic Indices and Syntactic Structures for Multilingual Question Answering: ITC-irst at CLEF 2005. In Peters C. (ed.): 6th Workshop of the Cross-Language Evaluation Forum, Vienna, Austria (2005).
12. Tapanainen P., Järvinen T.: A Non-Projective Dependency Parser. In Proceedings for the 5th Conference on Applied Natural Language Processing (1997).
13. Vicedo, J.L., Izquierdo R., Llopis F. and Muñoz R.: Question Answering in Spanish. In CLEF-2003 Workshop Notes, Trondheim, Norway (2003).