Towards Entailment-based Question Answering: ITC-irst at CLEF 2006

Milen Kouylekov, Matteo Negri, Bernardo Magnini and Bonaventura Coppola
Centro per la Ricerca Scientifica e Technologica ITC-irst
{kouylekov,magnini,negri,coppolab}@itc.it

Abstract

This year, besides providing support to other groups participating in cross-language Question Answering (QA) tasks, and submitting runs both for the monolingual Italian and the cross-language Italian/English tasks, the ITC-irst participation in the CLEF campaign concentrated on the Answer Validation Exercise (AVE). The participation in the AVE task, with an answer validation module based on textual entailment recognition, is motivated by our objectives of *(i)* creating a modular framework for an entailment-based approach to QA, and *(ii)* developing, in compliance with this framework, a stand-alone component for answer validation which implements different approaches to the problem.

Keywords

Question answering, Recognizing textual entailment, Tree edit distance, Answer validation

1 Introduction

This year the ITC-irst activity on Question Answering (QA) has been concentrated on two main directions: system re-engineering and entailment-based QA. The objective of the re-engineering activity is to transform our DIOGENE QA system into a more modular architecture whose basic components are completely separated, easily testable with rapid turnaround of controlled experiments, and easily replaceable with new implementations. In this direction, our long-term ambition is to make all these components (or even the entire system) freely available in the future to the QA community for research purposes.

Adhering to this perspective, the objective of the research on entailment-based QA aims at creating a stand-alone package for Answer Validation (AV), which implements different strategies to select the best answer among a set of possible candidates. Besides our well tested statistical approach to the problem, described in [11], we plan to include in this package a new AV method based on textual entailment recognition, allowing the user for experimentations with different AV settings.

Up to date we are halfway in this ambitious roadmap, and unfortunately the results of our participation in CLEF reflect this situation. While most of the original system's components are now separated from each other, little has been done to improve their performance, and a number of bugs still affect the overall system's behavior. The main improvements concern the substitution of the MG search engine with Lucene (an open-source, cross-platform search engine, which allows for customizable document ranking schemes), and the extension of the Answer Extraction component with a new WordNet-based module for handling "*generic*" factoid questions (*i.e.* questions whose expected answer type does not belong to the six broad named entity categories currently handled by DIOGENE). Both these improvements, and the results achieved by DIOGENE in the Monolingual Italian and cross-language Italian/English tasks are presented in Section 5.

As for the AV problem, even though the new component based on textual entailment recognition has not been integrated in the new DIOGENE architecture, it has been separately evaluated within the CLEF-2006 Answer Validation Exercise (AVE). The description of such component, its role in the upcoming new DIOGENE architecture, together with the results achieved in the AVE task represent the focus of this paper (Sections 2, and 3, and 4).

ITC-irst also provided support to the University "A.I.Cuza" of Iasi, Romania, involved in the cross-language Romanian/English QA task. Such support (concerning the document retrieval, answer extraction, and answer validation phases) is not documented in the paper as it is based on the same components used for our QA submissions.

2 Answer Validation in the QA loop

Answer validation is a crucial step in the Question Answering loop. After a *question processing* phase, where a natural language question is analyzed in order to extract all the information (*e.g.* relevant keywords, the expected answer type) necessary for the following steps of the process, a *passage retrieval* phase (where a ranked list of relevant passages is extracted from a target collection), and an *answer extraction* phase (where possible answers are extracted from the retrieved passages), QA systems are usually required to rank a large number of answer candidates, and finally select the best one.

The problem of answer ranking/validation has been addressed in different ways, ranging from the use of *answer patterns*, to the adoption of *statistical techniques* and other *type checking* approaches based on semantics.

Pattern-based techniques (see for example [7], and [16]) are based on the use of surface patterns (regular expressions, either manually created or automatically extracted from the Web) in order to model the likelihood of answers given the question.

Statistical techniques [11] are based on computing the probability of co-occurrence of the question terms and a candidate answer either on the Web, or in a local document collection.

Semantic type checking [15] techniques exploit structured and semi-structured data sources to determine the semantic type of suggested answers, in order to retain only those matching the expected answer type category.

Most of the proposed approaches, however, are not completely suitable to model all the linguistic phenomena that may be involved in the QA process. This particularly holds when the challenge moves from factoid questions (questions for which a complete answer can be given in a few words, usually corresponding to a named entity) to more complex ones (*e.g.* temporally restricted questions, "why questions", "how questions"). In order to capture the deeper semantic phenomena that are involved in determining answerhood, more complex techniques that approximate the forms of inference required to identify valid textual answers become necessary. In this direction, a more complex approach has been recently proposed by [4], which experiments with different techniques based on the recognition of *textual entailment* relations, either between questions and answers, or between questions and retrieved passages, to improve the accuracy of an open-domain QA system. The proposed experiments demonstrate that considerable gains in performance can be obtained by incorporating a textual entailment recognition module both during answer processing (+11%, in terms of Mean Reciprocal Rank, over the baseline system which does not include any entailment recognition processor) and passage retrieval (+25% MRR).

Focusing on the answer validation problem, our work builds on the same hypothesis that recognizing a textual entailment relation between a question and its answer can enhance the whole QA process. The following Sections provide an overview of the problem, together with a description of the component designed for the participation to the AVE task at CLEF 2006.

3 Entailment Recognition and Answer Validation

While the language variability problem is well known in Computational Linguistics, a general unifying framework has been proposed only recently in [2]. In this framework, language variability is addressed by defining the notion of *textual entailment* as a relation that holds between two language expressions (*i.e.* a *text* T and an *hypothesis* H) if the meaning of H, as interpreted in the context of T, can be inferred from the meaning of T. The entailment relation is directional, as the meaning of one expression can entail the meaning of the other, while the opposite may not.

Recently, the task of automatically Recognizing Textual Entailment (RTE) attracted a considerable attention and has been included among the tasks of the PASCAL Challenge [3]. Given two text fragments (T and H), systems participating to the PASCAL-RTE Challenge are required to automatically determine weather an entailment relation holds between them or not. The view underlying the RTE challenge [3] is that different natural language processing applications, including Question Answering (QA), Information Extraction (IE), (multi-document) summarization, and Machine Translation (MT), have to address the language variability problem and would benefit from textual entailment in order to recognize that a particular target meaning can be inferred from different text variants.

As the following examples illustrate, the task potentially covers almost all the phenomena involved in language variability. Entailment can in fact be due to lexical variations (example 1), syntactic variations (example 2), semantic inferences (example 3), or complex combinations of all these levels.

1. *T* – Euro-Scandinavian media cheer Denmark v Sweden draw.
   H – Denmark and Sweden tie.

2. *T* – Jennifer Hawkins is the 21-year-old beauty queen from Australia.
   H – Jennifer Hawkins is Australia's 21-year-old beauty queen.

3. *T* – The nomadic Raiders moved to LA in 1982 and won their third Super Bowl a year later.
   H – The nomadic Raiders won the Super Bowl in 1982.

Due to the variety of phenomena involved in language variability, one of the crucial aspects for any system addressing RTE is the amount of knowledge required for filling the gap between T and H. The basic inference technique shared by almost all the systems participating in PASCAL-RTE is the estimation of the degree of overlap between T and H. Such overlap is computed using a number of different approaches, ranging from statistic measures like *idf*, to deep syntactic processing and semantic reasoning. Some of them are relevant or similar to our methodology, as it will be described in the rest of the paper. For instance, the system described in [6] relies on dependency parsing and extracts lexical rules from WordNet. A decision tree based algorithm is used to separate the positive from the negative examples.

Similarly, in [1] the authors describe two systems for recognizing textual entailment. The first one is based on deep syntactic processing. Both T and H are parsed and converted into a logical form. An event-oriented statistical inference engine is used to separate the TRUE from FALSE pairs. The second system is based on statistical machine translation models.

Finally, a method for recognizing textual entailment based on graph matching is described in [13]. To handle language variability, the system uses a maximum entropy co-reference classifier and calculates term similarities using WordNet.

3.1 A distance-based approach for RTE

Our approach to RTE (fully described in [8], and [9]) is based on the intuition that, in order to discover an entailment relation between T and H, we must show that the whole content of H can be mapped into the content of T. The more straightforward the mapping can be established, the more probable is the entailment relation. Since a mapping can be described as the sequence of editing operations needed to transform T into H, where each edit operation has a cost associated

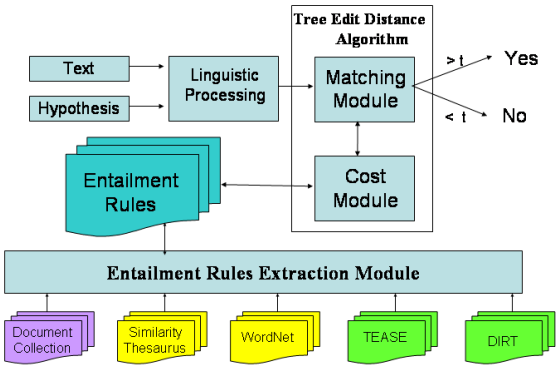

Figure 1: RTE System architecture

with it, we assign an entailment relation if the overall cost of the transformation is below a certain threshold, empirically estimated on the training data. For this cost estimation purpose, we adopted the tree edit distance algorithm described in [19], and applied it to the syntactic representations (*i.e.* dependency trees) of both T and H.

Edit operations are defined at the level of single nodes of the dependency tree (*i.e.* transformations on subtrees are not allowed in the current implementation). Since the tree edit distance algorithm adopted does not consider labels on edges, while dependency trees provide them, each dependency relation R from a node A to a node B has been re-written as a complex label B-R concatenating the name of the destination node and the name of the relation. All nodes, except the root of the tree, are re-labeled in such way. The algorithm is directional: we aim to find the better (*i.e.* less costly) sequence of edit operation that transform T (also called the *source*) into H (the *target*). According to the constraints described above, the following transformations are allowed:

- **Insertion**: insert a node from the dependency tree of H into the dependency tree of T. When a node is inserted it is attached with the dependency relation of the source label.
- **Deletion**: delete a node N from the dependency tree of T. When N is deleted all its children are attached to the parent of N. It is not required to explicitly delete the children of N as they are going to be either deleted or substituted on a following step.
- **Substitution**: change the label of a node N1 in the source tree into a label of a node N2 of the target tree. Substitution is allowed only if the two nodes share the same part-of-speech. In case of substitution the relation attached to the substituted node is changed with the relation of the new node.

3.1.1 RTE System Architecture

Our RTE system is composed by the following modules, showed in Figure 1: *(i)* a *text processing* module, for the preprocessing of the input T/H pair; *(ii)* a *matching module*, which performs the mapping between T and H; *(iii)* a *cost module*, which computes the cost of the edit operations.

The **text processing module** creates a syntactic representation of a T/H pair and relies on a sentence splitter and a syntactic parser. For sentence splitting we used *MXTerm* [14], a Maximum entropy sentence splitter. For parsing we used *Minipar*, a principle-based English parser [10] which has high processing speed and good precision.

The **matching module**, which implements the edit distance algorithm described in Section 3.1, finds the best sequence of edit operations between the dependency trees obtained from T and H. The entailment *score* of a given pair is calculated in the following way:

$$score(T, H) = \frac{ed(T, H)}{ed(, H)} \quad (1)$$

where $ed(T, H)$ is the function that calculates the edit distance cost and $ed(, H)$ is the cost of inserting the entire tree H. A similar approach is presented in [12], where the entailment score of two documents d and $d^{'}$ is calculated by comparing the sum of the weights (idf) of the terms that appear in both the documents to the sum of the weights of all terms in $d^{'}$. We used a threshold t such that if $score(T, H) < t$ then T entails H, otherwise no entailment relation holds for the pair. To set the threshold we have used both the positive and negative examples of the training set provided by the PASCAL-RTE dataset.

The matching module makes requests to the **cost module** in order to receive the cost of each single edit operation needed to transform T into H. For this purpose, we use different cost estimation strategies, based on the knowledge acquired from different linguistic resources. In particular:

- **Insertion.** The intuition underlying insertion is that its cost is proportional to the relevance of the word w to be inserted. We measure the relevance in terms of information value (*inverse document frequency (idf)*), position in the dependency tree, and ambiguity (in terms of number of WordNet senses).
- **Deletion.** Deleted words influence the meaning of already matched words. This requires that the evaluation of the cost of a deleted word is done after the matching is finished, with measures for estimating co-occurence (*e.g.* Mutual Information).
- **Substitution.** The cost of substituting a word w_1 with a word w_2 can be estimated considering the semantic entailment between the words. The more the two words are entailed, the less the cost of substituting one word with the other. We have defined a set of entailment rules over the WordNet relations among synsets, with their respective probabilities. If A and B are synsets in WordNet 2.0, then we derived an entailment rule in the following cases: A is hypernym of B; A is synonym of B; A entails B; A pertains to B.

The difficulty of the RTE task explains the relatively poor performance of the systems participating in the PASCAL Challenge. Most of them, in fact, achieved accuracy results between 52-58% at PASCAL1 and 55-65% at PASCAL2, with a random baseline of 50%. Even though our system's performance is in line with these results, (55% at PASCAL 1 and 60.5% at PASCAL 2), our 60% accuracy on the QA pairs in the PASCAL2 test set demonstrates the potentialities of integrating RTE in a QA system, and motivated our participation in the CLEF-2006 AVE task.

3.2 RTE for Answer Validation

In terms of the approach to RTE previously described, the answer validation task can be attacked as follows:

> A candidate answer *ca* to a question q is a *correct answer* if and only if the text from which *ca* is extracted entails the affirmative form of q, with *ca* inserted into the appropriate position.

As an example, consider the question/answer pair:
Q:"*Which country did Iraq invade in 1990?*"
A:"*Kuwait*"
In our approach, the solution to the problem consists in determining if the text T from which *ca* is extracted (for instance: "*The ship steamed into the Red Sea not long after Iraq invaded Kuwait in 1990.*"), entails the affirmative form H of q (*i.e.* "*Iraq invaded the country of Kuwait in 1990.*").

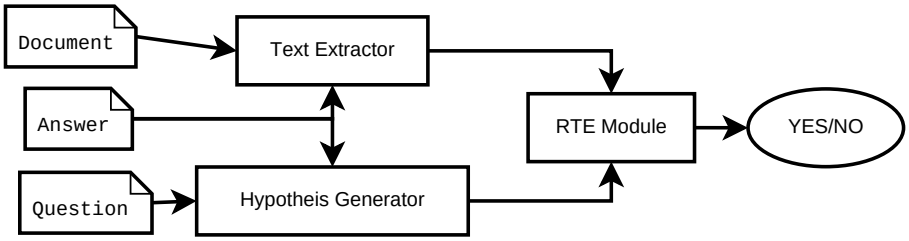

Figure 2: AV module architecture

A possible architecture of an answer validation component based on such RTE-based approach is depicted in Figure 3.2. In such architecture, the edit distance algorithm is launched over the output of two pre-processing modules, namely the *Text Extractor*, and the *Hypothesis Generator*.

The former, in charge of producing T, extracts from the document containing the candidate answer several sentences that will serve as input to the RTE module. As normally only one of these sentences contains the answer, the Text Extractor should be able to deal with different discourse phenomena (such as anaphora, ellipses, etc.) before constructing the final H to be passed to the RTE module.

The latter, in charge of producing H, transforms the input question into its affirmative form and inserts the answer on the proper position. To this aim, we developed a rule-based approach which addresses the problem by converting the dependency tree of the question into a syntactic template [17].

Finally, the RTE module produces a YES/NO answer, together with a certain confidence score.

4 Participation in the AVE task

As the PASCAL and AVE evaluation scenarios are slightly different, the datasets provided by the organizers of the two challenges also differ. In particular, the input text portions (candidate answers) contained in the AVE dataset are *document fragments* extracted from the participants' submissions to the CLEF QA track. Under these conditions, the complexity of the AVE task is undoubtedly higher due to the fact that: *(i)* there are no warranties about the syntactic correctness of those fragments; *(ii)* there are no constraints about their length, and *(iii)* the answer itself is not explicitly marked within them. As a consequence, under the RTE-based framework proposed in the previous sections, the availability of a module for transforming the text portions (candidate answers) into valid T inputs for the RTE module becomes crucial.

Unfortunately we were not able to complete this module (the *Text Extractor* in Figure 3.2) in time for the AVE submission deadline. In order to submit a run, we used the text portions provided by the organizers as Ts without any processing. As a result, we were not able to process the 129 pairs that actually didn't contain text (but only document identifiers), and we assigned them a NO response in the final submission.

The results achieved by our system are reported in Table 1. These results are not satisfactory, as they they approximate the behavior of a random approach. As stated before, a huge drawback for our system is the missing module for converting the document fragments into valid Ts for the RTE component. The absence of such module led to a significant drop in the performance of the parser, which is a crucial component in our RTE-based architecture. With these circumstances we can not draw definitive conclusions about the performance of our system, which requires at least a grammatically correct input. We assume that introducing the missing module we will significantly improve the performance of the system, allowing to reach results comparable with the ones obtained in the PASCAL 1 & 2 challenges.

Precision(YES)	Recall(YES)	F measure
0.3025	0.5023	0.3776

Table 1: AVE results

5 Participation in the QA Task: towards a new DIOGENE system

Our participation in the CLEF-2006 QA task is the first step on the way of developing a new and improved DIOGENE system. The leading principle of this re-engineering activity is to create a modular architecture, open to the insertion/substitution of new components. Another long-term objective of our work on QA, is to make the core components of the system freely available to the QA community for research purposes. In this direction, new implementation guidelines have been adopted, starting from the substitution of Lisp with Java as the main programming language. As stated before, our choices we were led by the following objectives:

1. Modularity - Re-build the old monolithic system into a pipeline of modules which share common I/O formats (xml files).

2. Configurable - Allow for the capability of configuring the settings of the different modules with external configuration files. We have developed a common xml schema for configuration description.

3. Evaluation - Provide the capability of performing fine-grained evaluation cycles over the individual processing modules which compose a QA system. In order to make a step-to-step evaluation of our modules we started to build reference datasets. Having such, the results of each processing step can be compared against them. At this stage, we focus on factoid questions from the TREC-2002 and TREC-2003 testsets. Exploiting such data, for instance, we are now able to perform separate tests on the Document Retrieval and the Candidate Answer Extraction steps. In the new architecture, evaluation modules are completely detached from processing modules. The former are independent from specific datasets and/or formats, enabling a quick porting along with the evolution of the task.

In spite of the huge effort in the modularity direction (now all the system's modules have been completely detached), little has been done to improve the behavior of the core components. The only improvements over the system described in [17] concern the *document retrieval* and *answer extraction* phases.

As for **document retrieval**, this year's novelty is the substitution of the MG search engine [18] with Lucene [5]. The many advantages of Lucene now do enable us to implement more refined strategies for indexing, ranking, and querying the underlying text collection. The Document Retrieval Evaluation is currently performed against the previous TREC/QA test sets, for which correct results are available. This only allows for a rough evaluation, since NIST results only report actual answers given by competing systems. However, for our goals such collections provide a reasonably fair coverage of correct answers. We perform Document Retrieval Evaluation at two different levels: Document Level and Text Passage Level. The first aims at evaluating the system against the official task, in which only the document reference must be provided. Conversely, Text Passage Level evaluation provides us with a better perspective on the following processing step, which is Candidate Answer Extraction. In fact, we fragment the AQUAINT corpus in paragraphs (our Text Passages) with respect to indexing. And, these paragraphs are exactly the text units over which answer extraction is performed.

As for **answer extraction**, we improved the candidate answers' selection process by developing a more effective way to handle "*generic*" factoid questions (*i.e.* questions whose expected answer type does not belong to the six broad named entity categories previously handled by DIOGENE).

Generic questions are now handled extracting as possible answer candidates all the hyponyms of the question focus that are found in the text passages returned by the search engine. For instance, given the question Q: *"What instrument did Jimi Hendrix play?"*, the answer extraction module will select *"guitar"* as a possible answer candidate as it is an hyponym of *"instrument"*, the focus of the question.

Apart from these modifications, the runs submitted to this year's edition of the CLEF QA task (results are reported in Table 2) have been produced with the old system's components, and reflect the "work-in-progress" situation of DIOGENE.

task	*Overall (%)*	*Def. (%)*	*Factoid (%)*	*Temp. (%)*
Italian/Italian	22.87	17.07	25.00	0.00
Italian/English	12.63	0.00	16.00	0.00

Table 2: System performance in the QA tasks

References

[1] Samuel Bayer, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. Mitre's submissions to the eu pascal rte challenge. In *Proceedings of PASCAL Workshop on Recognizing Textual Entailment*, Southampton, UK, 2005.

[2] Ido Dagan and Oren Glickman. Generic applied modeling of language variability. In *Proceedings of PASCAL Workshop on Learning Methods for Text Understanding and Mining*, Grenoble, 2004.

[3] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognizing textual entailment challenge. In *Proceedings of PASCAL Workshop on Recognizing Textual Entailment*, Southampton, UK, 2005.

[4] Sanda Harabagiu and Andrew Hickl. Methods of using textual entilment in open-domain question answering. In *In Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics, ACL-2006*, Sydney Australia, July 2006.

[5] Erik Hatcher and Otis Gospodnetic. *Lucene in Action (In Action series)*. Manning Publications, December 2004.

[6] Jesus Herrera, Anselmo Pe nas, and Felisa Verdejo. Textual entailment recognition based on dependency analysis and wordnet. In *Proceedings of PASCAL Workshop on Recognizing Textual Entailment*, Southampton, UK, 2005.

[7] E. Hovy, U. Hermjakob, and D. Ravichandran. A. Question/answer typology with surface text patterns. In *Proceedings of the DARPA Human Language Technology Conference (HLT)*, San Diego, CA, 2002.

[8] Milen Kouylekov and Bernardo Magnini. In *Proceedings of PASCAL Workshop on Recognizing Textual Entailment*, Southampton, UK, 2005.

[9] Milen Kouylekov and Bernardo Magnini. In *Proceedings of PASCAL Workshop on Recognizing Textual Entailment*, Venezia, UK, 2006.

[10] Dekang Lin. A maximum entropy part-of-speech tagger. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC-98*, Granada, Spain, 1998.

[11] Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Is it the right answer? exploiting web redundancy for answer validation. In *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL-2002*, pages 1495–1500, Philadelphia (PA), 7-12 July 2002.

[12] Christof Monz and Maarten de Rijke. Light-weight entailment checking for computational semantics. In *Proceedings of The third workshop on inference in computational semantics*, pages 59–72, Siena, Italy, 2001.

[13] Rajat Raina, Aria Haghighi, Christopher Cox, Jenny Finkel, Jeff Michels, Kristina Toutanova, Bill MacCartney, Marie-Catherine de Marneffe, Christopher D. Manning, and Andrew Y. Ng. Robust textual inference using diverse knowledge sources. In *Proceedings of PASCAL Workshop on Recognizing Textual Entailment*, Southampton, UK, 2005.

[14] Adwait Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceeding of the Empirical Methods in Natural Language Processing Conference*, University of Pennsylvania, 1996.

[15] S. Schlobach, M. Olsthoorn, and M. de Rijke. Type checking in open-domain question answering. In *"Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)"*, 2004.

[16] M. Subbotin. Patterns of potential answer expressions as clues to the right answers. In *Proceedings of the TREC-10 Conference*, pages 175–182, 2001.

[17] Hristo Tanev, MIlen Kouylekov, Matteo Negri, and Bernardo Magnini. Exploiting linguistic indices and syntactic structures for multilingual question answering: Itc-irst at clef 2005. In *CLEF-2005 Working*, Vienna, Austria, 2005.

[18] Ian H. Witten, Alistair Moffat, , and Timothy C. Bell. *The second edition of Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing.

[19] Kaizhong Zhang and Dennis Shasha. Fast algorithm for the unit cost editing distance between trees. In *Journal of algorithms, vol 11*, pages 1245–1262, Grenoble, 1990.