Question Answering over Structured Data: an Entailment-Based Approach to Question Analysis

Matteo Negri and Milen Kouylekov
FBK-irst - Fondazione Bruno Kessler
Via Sommarive 18, 38100 Povo (TN), Italy
negri,kouylekov@fbk.eu

Abstract

This paper addresses question analysis in the framework of Question Answering over structured data. The problem is set as a relation extraction task, where all the relations of interest in a given domain have to be extracted from natural language questions. The proposed approach applies the notion of Textual Entailment to compare the input questions with a repository of relational textual patterns. The underlying assumption is that a question expresses a certain relation if a pattern for that relation is entailed by the question. We report on a number of experiments, testing different simple distance-based entailment algorithms over a dataset of 1487 English questions covering the domain of cultural events in a town, and 75 relations that are relevant in this domain. The positive results obtained demonstrate the feasibility of the overall approach, and its effectiveness in the proposed QA scenario.

Keywords

Restricted-Domain Question Answering, Textual Entailment, Relation Extraction.

1 Introduction

Question analysis is the Question Answering (QA) subtask that consists in analysing a natural language question in order to identify all the relevant information needed to extract the correct answer from a given data source. Depending on the QA application, relevant information may include the identification of: *i)* the Expected Answer Type (*i.e.* the semantic category of the sought-after answer), *ii)* the word sense of the question terms, *iii)* the most important keywords, *iv)* named entities, and *v)* relations between entities. In the framework of QA over structured data, extracting from an input question all the relevant relations in a given domain becomes crucial, as it allows to fully capture the context in which the request has to be interpreted and, in turn, to determine the constraints on the database query. For instance, given the question *"What movie can I see today at cinema Astra?"*, an effective database query will select a concept of type MOVIE, with specific relations with a DATE (*e.g.* HASDATE(MOVIE:?, DATE:"today")) and a CINEMA (*e.g.* HASMOVIESITE(MOVIE:?, SITE:"Astra")). Successful answer retrieval depends on capturing *all and only* the relations expressed in the question: unrecognized relations will determine underspecified queries (often leading to redundant answers), while spuriously recognized relations will determine overspecified queries (leading to answer extraction failures).

While in open-domain QA any type of relation is potentially relevant, making their automatic identification unfeasible in an exhaustive manner, QA over structured data in a restricted domain presents a reasonable setting to address the task. In this paper we investigate the applicability of Textual Entailment (TE) as a possible solution to the problem. TE has been recently proposed as a comprehensive framework for applied semantics [4], where the mapping between linguistic objects is carried out by means of semantic inferences at the textual level. In the TE framework, a text (T) is said to entail the hypothesis (H) if the meaning of H can be derived from the meaning of T. According to such framework, we aim at discovering entailment relations between an input question Q (the *text* in the TE terminology) and a set of relational patterns (the *hypotheses*) that represent possible lexicalizations of the relations of interest in a given domain. The assumption is that, if an entailment relation holds between Q and a pattern p associated to a relation R_i, then R_i is among the relations expressed in Q.

In contrast with traditional approaches to QA over structured data, the proposed solution allows for a more flexible mapping between linguistic expressions (*e.g.* lexical items, syntactic structures) and data objects (*e.g.* concepts and relations in a knowledge base). This is because much of the machinery implied in such mapping, such as the construction of a logical form [1], [11], is not required in the TE framework, where inferences are performed at the textual level.

A TE-based approach to QA over structured data has been recently proposed in [9], which describes a system for the Italian language based on Linear Distance, a word-level TE Recognition (RTE) algorithm. Even though the preliminary results reported by the authors are encouraging, several aspects of the methodology have not been investigated, leaving room for more comprehensive evaluations. This paper represents a significant step forward, as it extensively addresses the following open issues: *i)* how do different RTE algorithms perform in the task of recognizing relevant relations in a dataset of domain-specific questions? *ii)* what is the impact on performance obtained by varying the number of available patterns associated to each relation?, *iii)* what is the impact of the TE-based approach to Relation Extraction (RE) on the overall performance of a QA system?, and *iv)* what is the relationship between the general TE Recognition task, as it is formulated within the Pascal-RTE Challenge [5], and the specific application scenario here proposed? By answering these questions, the main contribution of this work consists in providing exhaustive experiments to demonstrate that QA over structured data can be cast as an RTE-related problem.

The paper is organized as follows. Section 2 defines the TE-based RE task. Sections 3 and 4 describe our approach to RTE, and our experimental setting. Section 5 discusses evaluation results. Section 6 overviews related works, and Section 7 concludes the paper drawing final remarks.

2 Task Definition

We define the TE-based RE task as a classification problem, where a question Q annotated with entities has to be assigned to all the relations R_1,...,R_n it expresses, selected from a predefined set R. For instance, given the question *"What can I see [DATE: today] at cinema [SITE: Astra]?"*, the following relations represent the expected output of the system:
R1: HasMovieSite(Movie:?, Site:"Astra")
R2: HasDate(Movie:?, Date:"today")

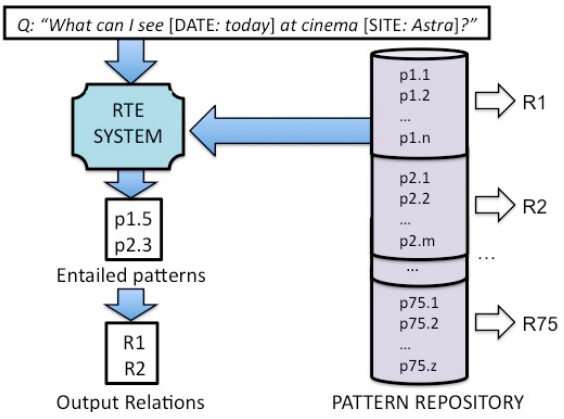

Fig. 1: *TE-based Relation Extraction process.*

As shown in Figure 1, the classification is carried out by means of an RTE system, which compares the question Q against a set of textual patterns stored in a *Pattern Repository (P)*. P contains n sets of relational patterns, each set representing possible lexicalizations of one relation R_i in R. Given the question Q, the RTE system attempts to verify, for each relation R_i, if an entailment relation holds between Q and at least one of R_i's patterns. If so, the relation is added to the output of the system. In case no relation is found, this is interpreted as evidence that the question is out of domain. Considering the example reported in Figure 1, since the input question entails patterns for the relations R1 and R2, the two relations are returned as output by the system. This task formulation is consistent with the one adopted in the Pascal-RTE initiative for the creation of IE pairs, as it is reported in [5].

2.1 Minimal Relational Patterns (MRPs)

Relational patterns represent an important aspect in our formulation of the task. In general, we say that a relational pattern p expresses a relation $R(arg1, arg2)$ in a certain language L if speakers of L agree that the meaning of p expresses the relation R between $arg1$ and $arg2$, given their knowledge about the entities. For instance, all the examples in Table 1 represent relational patterns for the relation HasMovieSite(Movie, Site).

In order to be profitably used in the proposed entailment framework, valid patterns should have the additional property of representing only *one* relation. Patterns representing multiple relations, in fact, would be entailed only by questions containing all those relations, thus resulting limited in their usage. For instance, only patterns (1)-(3) in Table 1 will be entailed by the question *"What can I see at cinema [SITE: Astra]?"*. Since (4) also contains the relation HasDate(Movie, Date), it will be entailed only by questions that lexicalize both relations.

Single-relation patterns, or *Minimal Relational Patterns* (MRPs), can be formally defined in terms of TE. Given two sets of relational patterns $P1$ and $P2$ for the relations $R1$ and $R2$, a pattern p_k belonging to $P1$ is a MRP for $R1$ if condition (1) holds.

$$\forall p_i \in P2, p_k \mapsto p_i = \emptyset \qquad (1)$$

In other words, a pattern p is minimal for a relation R if none of the patterns for the other relations can be derived from p (*i.e.* is entailed by p). According to such definition, patterns (1)-(3) are MRPs for the relation HasMovieSite(Movie, Site), while (4) is not, since it also entails patterns for the relation HasDate(Movie, Date).

3 Distance Based RTE

Edit distance approaches to RTE, such as the one proposed in [8], assume that the distance between T and H is a characteristic that separates the positive pairs, for which entailment holds, from the negative pairs, for which entailment does not hold. Such distance is computed as the cost of the editing operations (*i.e.* insertion, deletion and substitution) which are required to transform T into H. Each edit operation on two text fragments A and B (denoted as $A \rightarrow B$) has an associated cost (denoted as $\gamma(A \rightarrow B)$). The entailment score for a T-H pair is calculated on the minimal set of edit operations that transform T into H. An entailment relation is assigned to a T-H pair only if the overall cost of the transformation is below a certain threshold empirically estimated over training data. The entailment score function is defined in the following way:

$$score_{entailment}(T, H) = 1 - \frac{\gamma(T, H)}{\gamma_{nomap}(T, H)}$$

where $\gamma(T, H)$ is the function that calculates the edit distance between T and H, and $\gamma_{nomap}(T, H)$ is the *no mapping* distance equivalent to the cost of inserting

(1)	<ARG2:MOVIE:X> *is shown at cinema* <ARG1:CINEMA:Y>
(2)	*What* <ARG2:*movie*> *is on at* <ARG1:CINEMA:Y>*?*
(3)	*Is there any* <ARG2:*movie*> *that I can see at* <ARG1:CINEMA:Y>*?*
(4)	*Can I see* <ARG2:MOVIE:X> *at cinema* <ARG1:CINEMA:Y> *on* <ARG?:DATE:Z>*?*

Table 1: *Examples of relational patterns.*

the entire text of H, and deleting the entire text of T. The entailment score function has a range from 0 (when T is identical to H), to 1 (when T is completely different from H).

3.1 Algorithms

In this paper we experiment with the following two simple distance-based algorithms.

Linear Distance (LD) As for Linear Distance, *Levenshtein Distance* has been applied to RTE [8], by converting both the text T and the hypothesis H into sequences of words. Accordingly, edit operations have been defined as follows:

- **Insertion** ($\Lambda \rightarrow A$): insert a word A from H into T.
- **Deletion** ($A \rightarrow \Lambda$): delete a word A from T.
- **Substitution** ($A \rightarrow B$): substitute a word A from T with a word B from H.

Tree Edit Distance (TED) As regards Tree Edit Distance, [8] reports on an implementation for RTE based on [13], where the dependency trees of both T and H are considered. Edit operations are defined in the following way:

- **Insertion** ($\Lambda \rightarrow A$): insert a node A from the dependency tree of H into the dependency tree of T. When a node is inserted it is attached to the dependency relation of the source label.
- **Deletion** ($A \rightarrow \Lambda$): delete a node A from the dependency tree of T. When A is deleted all its children are attached to the parent of A. It is not required to explicitly delete the children of A, as they are going to be either deleted or substituted in a following step.
- **Substitution** ($A \rightarrow B$): change the label of a node A in the source tree into a label of a node B of the target tree. In case of substitution the relation attached to the substituted node is changed with the relation of the new node.

3.2 Cost Schemes for Edit Operations

The core of the edit distance approach is the mechanism for the definition of the cost of edit operations. This mechanism is defined separately from the distance algorithm and should reflect the knowledge of the user about the processed data. The principle behind it is to capture certain phenomena that facilitate the algorithm to assign small distances to positive T-H pairs, and large distances to negative pairs. Different semantic representations of the text allow different ways of defining the cost of edit operations. In the following paragraphs we describe the cost schemes we have used.

Default Cost Scheme (DEF)

$$\gamma(\Lambda \rightarrow A) = length(T)$$
$$\gamma(A \rightarrow \Lambda) = length(H)$$
$$\gamma(A \rightarrow B) = \begin{cases} 0 & A = B \\ \gamma_{i+d}(A \rightarrow B) & otherwise \end{cases}$$

In this scheme the cost of the insertion of a text fragment from H in T is equal to the length (*i.e.* the number of words) of T, and the deletion of a text fragment from T is equal to the length of H. The substitution cost is set to the sum of the insertion and the deletion of the text fragments, if they are not equal. This means that the algorithm would prefer to delete and insert text fragments rather than substituting them, in case they are not equal[1]. Setting the insertion and deletion costs respectively to the length of T and H is motivated by the fact that a shorter text T should not be preferred over a longer one T' while computing their overall mapping costs with the hypothesis H. Setting the costs to fixed values would in fact penalize longer texts (due to the larger amount of deletions needed) even though they are very similar to H.

When creating a cost scheme we can take advantage of other features of the processed text fragments. In the following two cost schemes we consider the depth and the width of the dependency trees representing the T-H pairs.

Depth-based Scheme (DS)

$$\gamma(\Lambda \rightarrow A) = depth(Tree_H) - depth(A)$$
$$\gamma(A \rightarrow \Lambda) = depth(Tree_T) - depth(A)$$

In this scheme the cost of the insertion of a node from the dependency tree of H in T, and of the deletion of a node from the dependency tree of T are inversely proportional to the depth (distance from the root) of the nodes in the dependency trees of T and H. The rationale behind this cost scheme is that words that are important to the meaning of the sentence, like verbs, subjects and objects, are usually in the top of the dependency tree and thus they should have higher costs of insertion and deletion.

[1] This is the default substitution setting for all the following schemes and will be omitted in their representation.

Width-based Scheme (WS)

$$\gamma(\Lambda \rightarrow A) = children(A)$$
$$\gamma(A \rightarrow \Lambda) = children(A)$$

In this scheme the cost of inserting and deleting a node is proportional to the number of children of the node in the dependency trees of T and H. The rationale is that the words that connect the phrases of the sentence are meaning preserving, and should have higher costs of insertion and deletion.

4 Experimental Setting

The main elements of our experimental setting are: *i)* the question corpus, and *ii)* the Pattern Repository.

4.1 Question Corpus

We experiment with a corpus of 1487 English questions extracted from the QALL-ME benchmark[2] [3], a multilingual corpus of annotated spoken requests in the domain of cultural events in a town (*e.g.* cinema, theatre, exhibitions, etc.). The available questions are manual transcriptions of *1223 read* and *264 spontaneous* telephone requests, annotated with different types of information. As far as relation annotation is concerned, questions are marked as containing one or more relations chosen from a set of 75 binary relations defined in the QALL-ME ontology[3]. As an example, the annotation of the question *Q2536: "What is the name of the director of 007 Casino Royale, which is shown today at cinema Modena?"* contains three relations, namely:

HasDate(Movie,Date)
HasMovieSite(Movie,Site)
HasDirector(Movie,Director).

The annotated questions contain 2 relations on average (min 1, max 6). A Kappa value of 0.94 (*almost perfect agreement*) was measured for the agreement between two annotators over part of the dataset (150 questions), showing the reliability of the annotation.

The annotated questions are used to create the *training* and *test* sets for our experiments. For this purpose, the question corpus was randomly split in two sets, respectively containing 999 and 488 questions. Such separation was carried out guaranteeing that, for each relation R, the questions marked with R are distributed in the two sets in proportion 2/3-1/3. The larger set of 999 questions is used for the acquisition of MRPs and, together with the resulting Pattern Repository, is used to train our RTE system (*i.e.* to empirically estimate an entailment threshold for each relation, considering positive and negative examples). The smaller set of 488 questions (which remained "unseen" in the MRP acquisition phase) is used as test set for the experiments described in Section 5.

4.2 Pattern Repository

According to the definition in Section 2.1, for each relation R we manually[4] extracted a set of MRPs from the training questions annotated with R. Given Q, the set of all the questions annotated with R, we adopt the following pattern creation guidelines:

1. A valid MRP describes only one relation.
2. A valid MRP has to be entailed by all the questions in Q.

For instance, given the following training examples for the relation HasDirector(Movie,Director):

Q493: *"What is the title of the last action movie directed by Martin Campbell?"*
Q2056: *"Is Gabriele Muccino's movie La Ricerca Della Felicitá on tomorrow?"*
Q2893: *"What is the name of the director of dreamgirls today at Nuovo Roma cinema?"*

the extracted MRPs are:

p1: movie directed by [Person]
p2: [Person]'s movie
p3: director of [Movie]

Adopting the aforementioned criteria, we populated our Pattern Repository with a total of 449 patterns, with at least 1 MRP per relation (6 on average).

5 Experiments and Discussion

5.1 Comparison of Different Algorithms

The objective of our first experiment was to determine the impact on RE performance of different configurations of the RTE system. As a baseline we used the *Longest Common Subsequence* (**LCS**), a similarity measure often used by RTE systems [2]. Given a text $T =< t_1, ..., t_n >$, and a hypothesis $H =< h_1, ..., h_n >$, the LCS is defined as the longest possible sequence $W =< w_1, ..., w_n >$ with words in W also being words in T and H in the same order.

For a meaningful comparison, we considered the following combinations of distance algorithms and cost schemes described in Section 3:

- Linear Distance + Default Scheme (**LD+DEF**) - to compare this word-level algorithm with those based on syntactic structures matching;
- Tree Edit Distance + Dynamic Scheme (**TED+DEF**) - to evaluate the contribution of considering dependency tree representations of the T-H pairs (obtained using Minipar[5]);
- Tree Edit Distance + Depth Scheme (**TED+DS**);
- Tree Edit Distance + Width Scheme (**TED+WS**).

[2] The QALL-ME benchmark has been developed within the EU funded QALL-ME project (http://qallme.fbk.eu).

[3] The QALL-ME ontology contains 107 properties (relations), and 122 classes (concepts). Our 75 relations, selected from the 107 properties, involve 27 concepts.

[4] Even though automatic pattern acquisition (from local corpora or from the Web) is an active research area, this aspect falls beyond the scope of this paper, and is left for future work.

[5] http://www.cs.ualberta.ca/∼lindek/minipar.htm

	LCS	LD+DEF	TED+DEF	TED+DS	TED+WS	ALL	ALL+PS
Precision	0.557	0.724	0.687	0.693	0.802	0.832	0.860
Recall	0.233	0.521	0.470	0.468	0.501	0.592	0.633
F1	0.318	0.606	0.559	0.559	0.617	0.692	**0.729**

Table 2: *Performance of different configurations of the RTE system on the test set.*

The distances resulting from the previous configurations are also used as features to train a classifier with a Random Forest Learning algorithm[6], obtaining the last experimented configuration:

- **ALL** - to evaluate the potential of a combination of all distances.

Each configuration was trained on the training set (999 questions), and then run on the test set (488 questions). Table 2 reports the results achieved on the test set. Precision/Recall/F1 scores indicate the system's ability to recognize the relations expressed in the test questions. Considering these results, we can draw the following conclusions:

1. All the configurations of the system significantly outperform the baseline (LCS), showing that distance-based algorithms are more suitable to capture entailment relations than simple word matching techniques.

2. As far as the single distance-based algorithms are concerned, TED taken in isolation slightly improves over LD only in one case (TED+WS). In general, we observe that TED alone achieves lower recall than LD. This can be explained by the parser difficulties in processing questions, and handling some syntactic structures like conjunctions, appositions, and relative clauses. TED+WS performs better than the other TED configurations as it handles compound nouns and PP phrases more effectively (*i.e.* it assigns lower costs of deletion to words that connect the main verb with its complements). Consider, for instance, the following T-H pair:

Q7: *"Where can I see the movie* [Movie:*Shrek*]*?"*
p: *where can i see [*MOVIE*]*

In this case, to make the complete mapping between T and H, the edit distance algorithm has to delete from T the word *"movie"*, which is part of a compound noun phrase. Using TED+WS, the contribution of such edit operation to the overall entailment score of the T-H pair will be lower than in the other TED-based configurations.

3. In spite of a lower recall, TED+WS achieves higher precision than LD. This validates the hypothesis that, when T and H have similar structures, words with a higher number of children (*i.e.* those connecting the phrases of the sentence) are meaning preserving, and should have higher costs of insertion and deletion.

4. The best result, achieved by the combined configuration (ALL), demonstrates that the different combinations of distance algorithms and cost schemes cover different entailment phenomena, and together they improve over the baseline up to +117% (from 0.318 to 0.692 F1).

5. The combined configuration (ALL) achieves good results especially in terms of Precision. This is particularly important in view of the overall QA application, for which high precision is a requirement. The possibility of answering a question Q, in fact, depends on system's ability to avoid overspecified queries due to false positives in the RE phase (*i.e.* recognized relations that are not present in Q).

5.2 Pattern Selection

Another important aspect in our approach is the relation between the number of patterns available in the Pattern Repository, and the performance of the system under different configurations. On the one side, we could expect that the more the patterns, the less the workload of the RTE system. Under this hypothesis, larger amounts of patterns will increase the possibility of discovering entailment relations. On the other side, dealing with many patterns could affect system's performance, as they might reduce the distance between positive and negative examples for a given relation in the training phase. This happens, for instance, when one of the variants for a relation $R1$ has many words in common with a pattern for another relation $R2$. To investigate this aspect, a pattern selection process has been carried out to select, for each relation R, the subset of the available MRPs (from the *power set* P(S) of the patterns for R) with highest precision on the training set. The pattern selection process has been carried out for each system configuration, and evaluated on the test data.

The last column of Table 2 (**ALL+PS**) reports the highest result, achieved by the combined configuration. This result demonstrates the positive impact of the pattern selection process, with an F1 improvement of +5.34% (from 0.692 to 0.729). The performance increase in the combined configuration is due to improved F1 results under *all* the configurations (the F1 improvements for the single configurations, not reported in Table 2, range from +0.16% for TED+WS, to +6% for LD+DEF). The minimal increase achieved by TED+WS shows that, in general, such configuration makes a better use of the available patterns. Such conclusion is supported by Table 3, which reports the number of patterns discarded under each configuration. As can be seen, the pattern selection algorithm eliminates significantly more patterns for the LD-based configuration than for the TED-based ones. The discovered correlation between *i)* the variations in the number of patterns available, *ii)* the system's performance variations, and *iii)* the type of TE recognition algorithm used, shows that larger amounts of patterns can be profitably used only with more sophisticated (*i.e.* semantically oriented) algorithms.

[6] Implemented in Weka: http://www.cs.waikato.ac.nz/ml/weka/

LD+DEF	TED+DEF	TED+DS	TED+WS
74	48	48	46

Table 3: *Discarded MRPs in each system configuration.*

5.3 Extrinsic Evaluation: Impact on QA

The third experiment aims at estimating the impact of our TE-based approach to question analysis on the overall performance of a QA system. For this purpose, each relation R in the Pattern Repository (P) can be associated to an SQL query to the database. The idea is that the system will first try to establish an entailment relation between an input question and each of the MRPs in P. Then, the SQL queries associated to the relations for which entailed patterns have been found will be joined in a single query. Our assumption is that effective database queries depend on recognizing all and only the relevant domain relations expressed in a question. As shown in the example below (referring to a question Q expressing the relations $R1$, $R2$, and $R3$), four types of queries can be obtained depending on the output of the RE phase:

- Type 1 - $[R1, R2, R3]$. Optimal case: the question analysis component correctly recognized all and only the relations expressed in Q. The conjunction of the SQL query portions associated to the three relations will correctly constraint the query, allowing for exact answer retrieval.
- Type 2 - $[R1, R2]$. Underspecified query: the missing constraint (*i.e.* the SQL query associated to $R3$) will lead to answers in which the sought-after information might come with non-relevant information[7].
- Type 3 - $[R1, R2, R3, R4]$. Overspecified query: the conjunction of a spurious SQL query portion (associated to $R4$) will lead to an answer extraction failure.
- Type 4 - $[R1, R2, R4]$. Mixed situation, leading to an answer extraction failure.

Table 4 reports the distribution of the four query types over the 488 test questions, obtained by the best configuration of the system (ALL), with and without pattern selection. Such distribution reflects the high precision of our TE-based question analysis component, especially when pattern selection is applied. In this case, around 36.5% of the test questions (178 out of 488) fall in the optimal case and, more important, around 83% of the questions (408) fall in the first two types (which at least lead to answers containing the sought-after information). As far as pattern selection is concerned, it's worth noting how its contribution comes both in terms of more Type 1 queries, and in terms of less Type 4 queries.

	Type 1	Type 2	Type 3	Type 4
ALL	164	232	20	72
ALL+PS	178	230	29	51

Table 4: *Query types distribution over 488 test questions.*

5.4 Evaluation over the RTE-3 Dataset

Our final experiment aims at better understanding the relationship between the general RTE task, as it is formulated within the Pascal-RTE Challenge, and the TE-based RE task here proposed. To compare the complexity of the two tasks the best configuration of our RTE system (ALL) has been trained and evaluated also on the RTE-3 dataset [6][8]. The resulting *63%* Accuracy roughly corresponds to the average performance of the systems participating in the challenge. Even though the two datasets are not comparable, the positive results achieved in both the evaluations demonstrate that systems designed for the general RTE task are perfectly suitable to address the problems posed by our application-oriented scenario.

6 Related Work

This section overviews related works, focusing on the differences between our approach and other TE-based approaches to QA and RE.

Question Answering. Several recent works document the use of TE as a mechanism for approximating the types of inference needed for QA. However, the QA subtasks addressed up to date (answer validation and ranking) differ completely from the problem discussed in the present work (question analysis). For instance, both in the Pascal-RTE Challenge, and in the CLEF-AVE task [10], the QA problem is modeled considering a question Q turned into an affirmative sentence as the hypothesis, and a text passage containing a candidate answer A as the text (*i.e.* systems have to decide whether A supports, or *entails*, Q). The same perspective is also adopted in [7], where TE is applied to filter and rank candidate answers returned by a QA system. While the application of TE for extracting relations in a given question is not documented in the QA field, similarities with our approach can be found in the RE area.

Relation Extraction. The most similar approach based on TE is described in [12], which reports experiments on a dataset of protein interactions. In spite of the similarities (*i.e.* the use of entailing templates for RE, and a syntax-based entailment checking), this approach differs from ours in several aspects. First, while [12] deals with a single relation, we consider a large number of possible target relations (*i.e.* 75), assuming that more than one relation can appear in a given question at the same time. Second, while [12]

[7] This, however, is a quite strong assumption. In some cases, in fact, missing relations do not affect the output of the system (*e.g.* given *"Who is the director of Casino Royale, today at Astra?"*, missing HasDate(Movie,Date), or HasMovieSite(Movie,Site), will not affect answer retrieval.)

[8] The Pascal-RTE3 dataset consists of 800 T-H pairs (the development set, which was used for training), and 800 T-H pairs (the test set, which was used for test).

deals with only one type of entities (*i.e.* proteins), in our multiple-relations scenario up to 27 entity types can participate in different relations. Finally, in [12] both the entities involved in the relation are given *a priori*, and the system has to decide whether, given two entities, they are involved in the relation or not. This assumption is not valid in our scenario, since it is not guaranteed that both the entities involved in a relation will appear in a given question.

7 Conclusions and Future Work

This paper addressed question analysis in the framework of QA over structured data, focusing on the task of extracting relations from a natural language question. We approached the problem by applying the notion of Textual Entailment to compare the input question with a repository of patterns representing different lexicalizations of the relevant relations in a given domain. The reported experiments demonstrate: *i)* the feasibility of the approach, *ii)* the correlation between the number of available patterns and the performance of different RTE algorithms, *iii)* the positive impact of our approach on the overall performance of a QA system, and *iv)* the suitability of systems designed for the general RTE task for the proposed application-oriented scenario.

Showing that even basic (general-purpose) RTE algorithms are suitable to address the task, our results motivate further research, with improved algorithms, along the same direction. Future work will thus concentrate on improving QA performance with more semantically oriented RTE algorithms. For example, enhanced cost schemes should apply *entailment rules* considering different features of the terms involved in the transformations, such as their *semantic similarity* (*e.g.* lower substitution costs for synonyms), and their *weight* (*e.g.* the insertion cost of a term t will be proportional to the number of relations whose patterns contain t). A complementary research direction is the automatic acquisition of relational patterns from the available dataset of questions. This will enhance the scalability of our approach (*i.e.* the possibility to enlarge the set of relevant domain relations), and its portability across domains.

References

[1] I. Androutsopoulos, G. Ritchie, and P. Thanisch. Natural Language Interfaces to Databases – An Introduction. *Journal of Natural Language Engineering*, 1(1), 1995.

[2] W. Bosma and C. Callison-Burch. Paraphrase Substitution for Recognizing Textual Entailment. In *Revised Selected Papers of CLEF 2006*, 2006.

[3] E. Cabrio, B. Coppola, R. Gretter, M. Kouylekov, B. Magnini, and M. Negri. Question Answering Based Annotation for a Corpus of Spoken Requests. In *Proceedings of the Workshop on Semantic Representation of Spoken Language SRSL*07, Salamanca, Spain, 2007.

[4] I. Dagan and O. Glickman. Generic Applied Modeling of Language Variability. In *Proceedings of the PASCAL Workshop on Learning Methods for Text Understanding and Mining*, Grenoble, France, 2004.

[5] I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges (MLCW 2005)*, volume 3944 of *LNAI*. Springer-Verlag, 2006.

[6] D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL 2007 PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, 2007.

[7] S. Harabagiu and A. Hickl. Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of COLING/ACL 2006*, Sydney, Australia, 2006.

[8] M. Kouylekov and B. Magnini. Combining Lexical Resources with Tree Edit Distance for Recognizing Textual Entailment. In *Machine Learning Challenges (MLCW 2005)*, volume 3944 of *LNAI*. Springer-Verlag, 2006.

[9] M. Negri, M. Kouylekov, and B. Magnini. Detecting Expected Answer Relations through Textual Entailment. In *Proceedings of CICLing 2008*, Haifa, Israel, 2008.

[10] A. Penas, A. Rodrigo, V. Sama, and F. Verdejo. Overview of the Answer Validation Exercise 2006. In *Evaluation of Multilingual and Multi-modal Information Retrieval*, volume 4730 of *LNCS*. Springer-Verlag, 2006.

[11] A. Popescu, O. Etzioni, and H. Kautz. Towards a Theory of Natural Language Interfaces to Databases. In *Proceedings of the Conference on Intelligent User Interfaces*, Miami, Florida, 2003.

[12] L. Romano, M. Kouylekov, I. Szpektor, I. Dagan, and A. Lavelli. Investigating a Generic Paraphrase-based Approach for Relation Extraction. In *Proceedings of the 11th EACL Conference*, Trento, Italy, 2006.

[13] K. Zhang and D. Shasha. Fast Algorithm for the Unit Cost Editing Distance Between Trees. *Journal of Algorithms*, 11, December 1990.