

# Splitting Complex Temporal Questions for Question Answering systems\*

E. Saquete, P. Martínez-Barco, R. Muñoz, J.L. Vicedo

Grupo de investigación del Procesamiento del Lenguaje y Sistemas de Información.

Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.

Alicante, Spain

{stela,patricio,rafael,vicedo}@dlsi.ua.es

## Abstract

This paper presents a multi-layered Question Answering (Q.A.) architecture suitable for enhancing current Q.A. capabilities with the possibility of processing complex questions. That is, questions whose answer needs to be gathered from pieces of factual information scattered in different documents. Specifically, we have designed a layer oriented to process the different types of temporal questions. Complex temporal questions are first decomposed into simpler ones, according to the temporal relationships expressed in the original question.

In the same way, the answers of each simple question are re-composed, fulfilling the temporal restrictions of the original complex question.

Using this architecture, a Temporal Q.A. system has been developed.

In this paper, we focus on explaining the first part of the process: the decomposition of the complex questions. Furthermore, it has been evaluated with the TERQAS question corpus of 112 temporal questions. For the task of question splitting our system has performed, in terms of precision and recall, 85% and 71%, respectively.

## 1 Introduction

Question Answering could be defined as the process of computer-answering to precise or arbitrary questions formulated by users. Q.A. systems are especially useful to obtain a specific piece of information without the need of manually going through all the available documentation related to the topic.

Research in Question Answering mainly focuses on the treatment of *factual questions*. These require as an answer very specific items of data, such as dates, names of entities or quantities, e.g., “*What is the capital of Brazil?*”.

Temporal Q.A. is not a trivial task due to the complexity temporal questions may reach. Current operational Q.A. systems can deal with simple factual questions. That is, questions requiring to be answered with a date, e.g. “*When did Bob Marley die?*”. or questions that include simple temporal expressions in their formulation, e.g., “*Who won the U.S. Open in 1999?*”. Processing this sort of questions is usually performed by identifying explicit temporal expressions in questions and relevant documents, in order to gather the necessary information to answer the queries.

Even though, it seems necessary to emphasize that the system described in (Breck et al., 2000) is the only one also using implicit temporal expression recognition for Q.A. purposes. It does so by applying the temporal tagger developed by Mani and Wilsson (2000).

However, issues like addressing the temporal properties or the ordering of events in questions, remain beyond the scope of current Q.A. systems:

- “Who was spokesman of the Soviet Embassy in Baghdad *during* the invasion of Kuwait?”
- “Is Bill Clinton *currently* the President of the United States?”

This work presents a Question Answering system capable of answering complex temporal questions. This approach tries to imitate human behavior when responding this type of questions. For example, a human that wants to answer the question: “*Who was spokesman of the Soviet Embassy in Baghdad during the invasion of Kuwait?*” would follow this process:

1. First, he would decompose this question into two simpler ones: “*Who was spokesman of the Soviet Embassy in Baghdad?*” and “*When did the invasion of Kuwait occur?*”.
2. He would look for all the possible answers to the first simple question: “*Who was*

\* This paper has been supported by the Spanish government, projects FIT-150500-2002-244, FIT-150500-2002-416, TIC-2003-07158-C04-01 and TIC2000-0664-C02-02.

*spokesman of the Soviet Embassy in Baghdad?”.*

3. After that, he would look for the answer to the second simple question: “*When did the invasion of Kuwait occur?*”
4. Finally, he would give as a final answer one of the answers to the first question (if there is any), whose associated date stays within the period of dates implied by the answer to the second question. That is, he would obtain the final answer by discarding all answers to the simple questions which do not accomplish the restrictions imposed by the temporal signal provided by the original question (*during*).

Therefore, the treatment of complex question is based on the decomposition of these questions into simpler ones, to be resolved using conventional Question Answering systems. Answers to simple questions are used to build the answer to the original question.

This paper has been structured in the following fashion: first of all, section 2 presents our proposal of a taxonomy for temporal questions. Section 3 describes the general architecture of our temporal Q.A. system. Section 4 deepens into the first part of the system: the decomposition unit. Finally, the evaluation of the decomposition unit and some conclusions are shown.

## 2 Proposal of a Temporal Questions Taxonomy

Before explaining how to answer temporal questions, it is necessary to classify them, since the way to solve them will be different in each case. Our classification distinguishes first between simple questions and complex questions. We will consider as simple those questions that can be solved directly by a current General Purpose Question Answering system, since they are formed by a single event. On the other hand, we will consider as complex those questions that are formed by more than one event related by a temporal signal which establishes an order relation between these events.

### Simple Temporal Questions:

*Type 1: Single event temporal questions without temporal expression (TE).* This kind of questions are formed by a single event and can be directly resolved by a Q.A. System, without pre- or post-processing them. There are not temporal expressions in the question. Example: “*When did Jordan close the port of Aqaba to Kuwait?*”

*Type 2: Single event temporal questions with temporal expression.* There is a single event in the ques-

tion, but there are one or more temporal expressions that need to be recognized, resolved and annotated. Each piece of temporal information could help to search for an answer. Example: “*Who won the 1988 New Hampshire republican primary?*”. TE: 1988

### Complex Temporal Questions:

*Type 3: Multiple events temporal questions with temporal expression.* Questions that contain two or more events, related by a temporal signal. This signal establishes the order between the events in the question. Moreover, there are one or more temporal expressions in the question. These temporal expressions need to be recognized, resolved and annotated, and they introduce temporal constraints to the answers of the question. Example: “*What did George Bush do after the U.N. Security Council ordered a global embargo on trade with Iraq in August 90?*” In this example, the temporal signal is *after* and the temporal constraint is “*between 8/1/1990 and 8/31/1990*”. This question can be divided into the following ones:

- Q1: What did George Bush do?
- Q2: When the U.N. Security Council ordered a global embargo on trade with Iraq?

*Type 4: Multiple events temporal questions without temporal expression.* Questions that consist of two or more events, related by a temporal signal. This signal establishes the order between the events in the question. Example: “*What happened to world oil prices after the Iraqi annexation of Kuwait?*”. In this example, the temporal signal is *after* and the question would be decomposed into:

- Q1: What happened to world oil prices?
- Q2: When did the Iraqi “annexation” of Kuwait occur?

How to process each type will be explained in detail in the following sections.

## 3 Multi-layered Question-Answering System Architecture

Current Question Answering system architectures do not allow to process complex questions. That is, questions whose answer needs to be gathered from pieces of factual information that is scattered in a document or through different documents. In order to be able to process these complex questions, we propose a multi-layered architecture. This architecture increases the functionality of the current Question-Answering systems, allowing us to solve any type of temporal questions. Moreover, this system could be easily augmented with new layers to

cope with questions that need complex processing and are not temporal oriented.

Some examples of complex questions are:

- Temporal questions like “Where did Michael Milken study before going to the University of Pennsylvania?”. This kind of questions needs to use temporal information and event ordering to obtain the right answer.
- Script questions like “How do I assemble a bicycle?”. In these questions, the final answer is a set of ordered answers.
- Template-based questions like “Which are the main biographical data of Nelson Mandela?”. This question should be divided in a number of factual questions asking for different aspects of Nelson Mandela’s biography. Gathering their respective answers will make it possible to answer the original question.

These three types of question have in common the necessity of an additional processing in order to be solved. Our proposal to deal with them is to superpose an additional processing layer, one by each type, to a current General Purpose Question Answering system, as it is shown in Figure 1. This layer will perform the following steps:

- Decomposition of the question into simple events to generate simple questions (sub-questions) and the ordering of the sub-questions.
- Sending simple questions to a current General Purpose Question Answering system.
- Receiving the answers to the simple questions from the current General Purpose Question Answering system.
- Filtering and comparison between sub-answers to build the final complex answer.

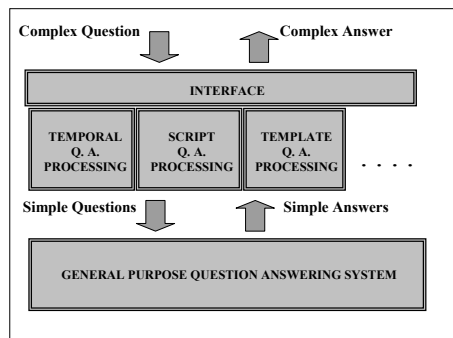


Figure 1: Multi-layered Architecture of a Q.A.

The main advantages of performing this multi-layered system are:

- It allows you to use any existing general Q.A. system, with the only effort of adapting the output of the processing layer to the type of input that the Q.A. system uses.
- Due to the fact that the process of complex questions is performed at an upper layer, it is not necessary to modify the Q.A. system when you want to deal with more complex questions.
- Each additional processing layer is independent from each other and only processes those questions within the type accepted by that layer.

Next, we present a layer oriented to process temporal questions according to the taxonomy shown in section 2.

### 3.1 Architecture of a Question Answering System applied to Temporality

The main components of the Temporal Question Answering System are (c.f. figure 2) top-down: Question Decomposition Unit, General purpose Q.A. system and Answer, Recomposition Unit.

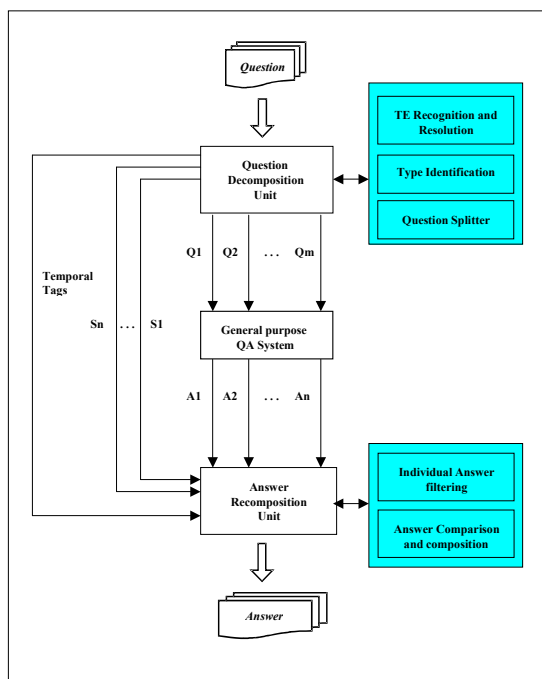


Figure 2: Temporal Question Answering System

These components work all together for the obtainment of a final answer. The Question Decomposition Unit and the Answer Recomposition Unit are the units that conform the Temporal Q.A. layer

which process the temporal questions, before and after using a General Purpose Q.A. system.

- *The Question Decomposition Unit* is a preprocessing unit which performs three main tasks. First of all, the recognition and resolution of temporal expressions in the question. Secondly, there are different types of questions, according to the taxonomy shown in section 2. Each type of them needs to be treated in a different manner. For this reason, type identification must be done. After that, complex questions of types 3 and 4 only, are split into simple ones, which are used as the input of a General Purpose Question-Answering system. For example, the question “*Where did Bill Clinton study before going to Oxford University?*”, is divided into two sub-questions related through the temporal signal *before*:

- Q1: Where did Bill Clinton study?
- Q2: When did *Bill Clinton* go to Oxford University?

- *A General Purpose Question Answering system*. Simple factual questions generated are processed by a General Purpose Question Answering system. Any Question Answering system could be used here. In this case, the SEMQA system (Vicedo and Ferrández, 2000) has been used. The only condition is to know the output format of the Q.A. system to accordingly adapt the layer interface. For the example above, a current Q.A. system returns the following answers:

- Q1 Answers: Georgetown University (1964-68) // Oxford University (1968-70) // Yale Law School (1970-73)
- Q2 Answer: 1968

- *The Answer Recomposition Unit* is the last stage in the process. This unit builds the answer to the original question from the answers to the sub-questions and the temporal information extracted from the questions (temporal signals or temporal expressions). As a result, the correct answer to the original question is returned.

Apart from proposing a taxonomy of temporal questions, we have presented a multi-layered Q.A. architecture suitable for enhancing current Q.A. capabilities with the possibility of adding new layers for processing different kinds of complex questions. Moreover, we

have proposed a specific layer oriented to process each type of temporal questions.

The final goal of this paper is to introduce and evaluate the first part of the temporal question processing layer: the Question Decomposition Unit.

Next section shows the different parts of the unit together with some examples of their behavior.

## 4 Question Decomposition Unit

The main task of this unit is the decomposition of the question, which is divided in three main tasks or modules:

- Type Identification (according to the taxonomy proposed in section 2)
- Temporal Expression Recognition and Resolution
- Question Splitter

These modules are fully explained below. Once the decomposition of the question has been made, the output of this unit is:

- A set of sub-questions, that are the input of the General Purpose Question-Answering system.
- Temporal tags, containing concrete dates returned by TERSEO system (Saquete et al., 2003), that are part of the input of the Answer Recomposition Unit and are used by this unit as temporal constraints in order to filter the individual answers.
- A set of temporal signals that are part of the input of the Answer Recomposition Unit as well, because this information is necessary in order to compose the final answer.

Once the decomposition has been made, the General Purpose Question-Answering system is used to treat with simple questions. The temporal information goes directly to the Answer Recomposition unit.

### 4.1 Type Identification

The Type Identification Unit classifies the question in one of the four types of the taxonomy proposed in section 2. This identification is necessary because each type of question causes a different behavior (scenario) in the system. Type 1 and Type 2 questions are classified as simple, and the answer can be obtained without splitting the original question. However, Type 3 and Type 4 questions need to be split in a set of simple sub-questions. The types of these sub-questions are always Type 1 or Type 2 or

a non-temporal question, which are considered simple questions.

The question type is established according to the rules in figure 3:

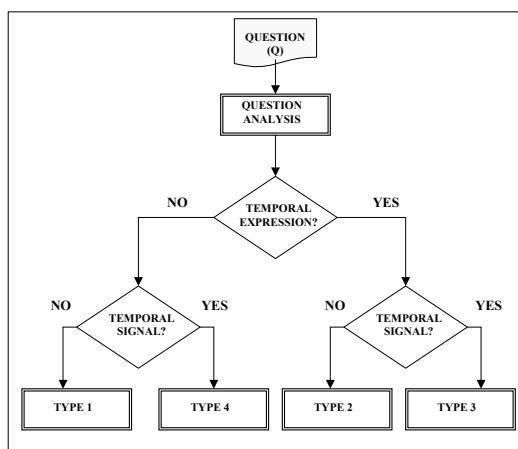


Figure 3: Decision tree for Type Identification

## 4.2 Temporal Expression Recognition and Resolution

This module uses TERSEO system (Saquete et al., 2003) to recognize, annotate and resolve temporal expressions in the question. The tags this module returns exhibit the following structure:

*Explicit dates:*

```
<DATE_TIME ID="value" TYPE="value"
VALDATE1="value" VALTIME1="value"
VALDATE2="value" VALTIME2="value">
expression </DATE_TIME>
```

*Implicit dates:*

```
<DATE_TIME_REF ID="value" TYPE="value"
VALDATE1="value" VALTIME1="value"
VALDATE2="value" VALTIME2="value">
expression </DATE_TIME_REF>
```

Every expression is identified by a numeric ID. VALDATE# and VALTIME# store the range of dates and times obtained from the system, where VALDATE2 and VALTIME2 are only used to establish ranges. Furthermore, VALTIME1 could be omitted if a single date is specified. VALDATE2, VALTIME1 and VALTIME2 are optional attributes.

These temporal tags are the output of this module and they are used in the Answer Recomposition Unit in order to filter the individual answers obtained by the General Purpose Question-Answering system. The tags are working as temporal constraints.

Following, a working example is introduced. Given the next question “Which U.S. ship was attacked by Israeli forces during the Six Day war in the sixties?”:

1. Firstly, the unit recognizes the temporal expression in the question, resolves and tags it, resulting in:

```
<DATETIMEREf valdate1="01/01/1960"
valdate2="31/12/1969"> in the
sixties </DATETIMEREf>
```

2. The temporal constraint is that the date of the answers should be between the values *valdate1* and *valdate2*.

## 4.3 Question Splitter

This task is only necessary when the type of the question, obtained by the Type Identification Module, is 3 or 4. These questions are considered complex questions and need to be divided into simple ones (Type 1, Type 2). The decomposition of a complex question is based on the identification of temporal signals, which relate simple events in the question and establish an order between the answers of the sub-questions. Finally, these signals are the output of this module and are described in next subsection.

### 4.3.1 Temporal Signals

Temporal signals denote the relationship between the dates of the related events. Assuming that *F1* is the date related to the first event in the question and *F2* is the date related to the second event, the signal will establish an order between them. This we have named the *ordering key*. An example of some ordering keys is introduced in table 1.

SIGNAL	ORDERING KEY
After	F1 > F2
When	F1 = F2
Before	F1 < F2
During	F2i <= F1 <= F2f
From F2 to F3	F2 <= F1 <= F3
About F2 -- F3	F2 <= F1 <= F3
On / in	F1 = F2
While	F2i <= F1 <= F2f
For	F2i <= F1 <= F2f
At the time of	F1 = F2
Since	F1 > F2

Table 1: Example of signals and ordering keys

### 4.3.2 Implementation

One have divided each complex question into two parts, based on the temporal signal. The former is a simple question, therefore, no transformation is required. However, the latter (the bit after the temporal signal) needs transformation into a correct question pattern, always corresponding to a “When” type-question. Moreover, three different kinds of question structures have been determined, being the transformation different for each of them. The implementation of this module is shown in figure 4.

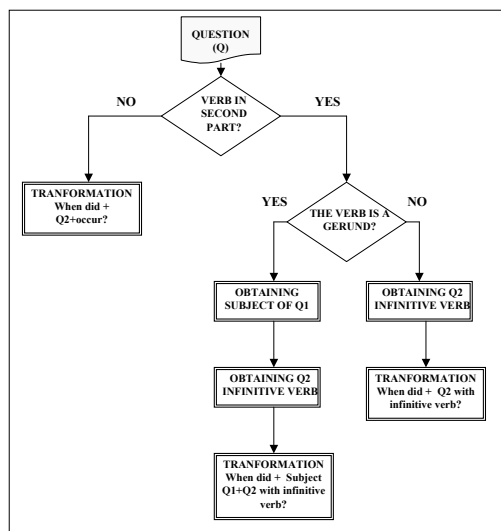


Figure 4: Decision tree for the Question Splitter

The three possible cases are:

- The question that follows the temporal signal does not contain any verb, for example: “*What happened to the world oil prices after the Iraqi annexation of Kuwait?*” In this case, our system returns the following transformation: “*When did the Iraqi annexation of Kuwait occur?*” This case is the simplest, since the only transformation needed is adding the words “When did... occur?” to the second sentence.
- The question that follows the temporal signal contains a verb, but this verb is a gerund tense, for example: “*Where did Bill Clinton study before going to Oxford University?*” In this case two previous steps to the transformation are necessary:
  1. Extracting the subject of the previous question.
  2. Converting the verb of the second sentence to infinitive tense.

The final question returned by the system is: “*When did Bill Clinton go to Oxford University?*”.

- In the last type of transformation the second sentence in the question contains a tensed verb and its own subject, e.g., “*What did George Bush do after the U.N. Security Council ordered a global embargo on trade with Iraq?*” In this case, the infinitive and the tense of the sentence are obtained. Hence, the question results in the following form: “*When did the U.N. Security Council order a global embargo on trade with Iraq?*”.

### 4.3.3 Example

In the following example a part of the returned file of our Decomposition Unit is shown.

1. Where did Bill Clinton study before going to Oxford University?  
Temporal Signal: before  
Q1: Where did Bill Clinton study?  
Q2: When did Bill Clinton go to Oxford University?
2. What did George Bush do after the U.N. Security Council ordered a global embargo on trade with Iraq in August 90?  
Temporal Signal: after  
Temporal Expression: in August 90  
Q1: What did George Bush do?  
Q2: When did the U.N. Security Council order a global embargo on trade with Iraq in August 90?  
DateQ2:[01/08/1990--31/08/1990]
3. When did Iraq invade Kuwait?  
Temporal Signal: -  
Temporal Expression: -  
Q1: When did Iraq invade Kuwait?
4. Who became governor of New Hampshire in 1949?  
Temporal Signal: -  
Temporal Expression: in 1949  
Q1: Who became governor of New Hampshire in 1949?  
DateQ1:[01/01/1949--31/12/1949]

### 4.4 Decomposition Unit Evaluation

This section presents an evaluation of the Decomposition Unit for the treatment of complex questions. For the evaluation a corpus of questions containing as many simple as complex questions is required. Due to the fact that question corpora used in TREC (TREC, ) and CLEF (CLEF, ) do not contain complex questions, the TERQAS question corpus has been chosen (Radev and Sundheim, 2002; Pustejovsky, 2002). It consists of 123 temporal questions.

	TOTAL	TREATED	SUCSESSES	PRECISION	RECALL	F-MEASURE
<b>TE Recognition and Resolution</b>	62	52	47	90%	75%	86%
<b>Type Identification</b>	112	112	104	92%	100%	93%
<b>Signal Detection</b>	17	14	14	100%	82%	95%
<b>Question Splitter</b>	17	14	12	85%	71%	81%
<b>DECOMPOSITION UNIT</b>	112	112	93	83%	83%	83%

Table 2: Evaluation of the system

From these, 11 were discarded due to requiring the need of a treatment beyond the capabilities of the system introduced hereby. Questions of the type: “Who was the second man on the moon” can not be answered by applying the question decomposition. They need a special treatment. For the aforementioned phrase, this would consist of obtaining the names of all the men having been on the moon, ordering the dates and picking the second in the ordered list of names.

Therefore, for this evaluation, we have just been focusing on trying to resolve the 112 left. The evaluation has been made manually by three annotators. Four different aspects of the unit have been considered:

- *Recognition and resolution of Temporal Expressions:* In this corpus, there were 62 temporal expressions and our system was able to recognize 52, from which 47 were properly resolved by this module.
- *Type Identification:* There were 112 temporal questions in the corpus. Each of them was processed by the module, resulting in 104 properly identified according to the taxonomy proposed in section 2.
- *Signal Detection:* In the corpus, there were 17 questions that were considered complex (Type 3 and Type 4). Our system was able to treat and recognize correctly the temporal signal of 14 of these questions.
- *Question Splitter:* From this set of 17 complex questions, the system was able to process 14 questions and divided properly 12 of them.

The results, in terms of precision and recall are shown in Table 2. In the evaluation, only 19 questions are wrongly pre-processed. Errors provoking a wrong pre-processing have been analyzed thoroughly:

- There were 8 errors in the identification of the type of the question and they were due to:
  - Not treated TE or wrong TE recognition: 6 questions.
  - Wrong Temporal Signal detection: 2 questions.
- There were 5 errors in the Question Splitter module:
  - Wrong Temporal Signal detection: 3 questions.
  - Syntactic parser problems: 2 questions.
- There were 15 errors not affecting the treatment of the question by the General Purpose Question Answering system. Nevertheless, they do affect the recomposition of the final answer. They are due to:
  - Not treated TE or wrong TE recognition: 6 questions.
  - Wrong temporal expression resolution: 9 questions.

Some of these questions provoke more than one problem, causing that both, type identification and division turn to be wrong.

## 5 Conclusions

This paper presents a new and intuitive method for answering complex temporal questions using an embedded current factual-based Q.A. system. The method proposed is based on a new procedure for the decomposition of temporal questions, where complex questions are divided into simpler ones by means of the detection of temporal signals. The TERSEO system, a temporal information extraction system applied to event ordering has been used to detect and resolve temporal expressions in questions and answers.

Moreover, this work proposes a new multi-layered architecture that enables to solve complex questions by enhancing current Q.A. capabilities.

The multi-layered approach can be applied to any kind of complex questions that allow question decomposition such as script questions, e.g., “*How do I assemble a bicycle?*”, or template-like questions, e.g., “*Which are the main biographical data of Nelson Mandela?*”.

This paper has specifically focused on a process of decomposition of complex temporal questions and on its evaluation on a temporal question corpus.

In the future, our work is directed to fine tune this system and increase its capabilities towards processing questions of higher complexity.

## References

- E. Breck, J. Burger, L. Ferro, W. Greiff, M. Light, I. Mani, and J. Rennie. 2000. Another sys called quanda. In *Ninth Text REtrieval Conference*, volume 500-249 of *NIST Special Publication*, pages 369–378, Gaithersburg, USA, nov. National Institute of Standards and Technology.
- CLEF. Cross-language evaluation forum. <http://clef.iei.pi.cnr.it/>.
- I. Mani and G. Wilson. 2000. Robust temporal processing of news. In ACL, editor, *Proceedings of the 38th Meeting of the Association of Computational Linguistics (ACL 2000)*, Hong Kong, October.
- J. Pustejovsky. 2002. Terqas:time and event recognition for question answering systems. <http://time2002.org/>.
- D. Radev and B. Sundheim. 2002. Using timeml in question answering. <http://www.cs.brandeis.edu/~jamesp/arda/time/documentation/TimeML-use-in-qa-v1.0.pdf>.
- E. Saquete, R. Muñoz, and P. Martínez-Barco. 2003. Terseo: Temporal expression resolution system applied to event ordering. In TSD, editor, *Proceedings of the 6th International Conference ,TSD 2003, Text, Speech and Dialogue*, pages 220–228, Ceske Budejovice,Czech Republic, September.
- TREC. Text retrieval conference. <http://trec.nist.gov/>.
- J. L. Vicedo and A. Ferrández. 2000. A semantic approach to question answering systems. In *Ninth Text REtrieval Conference*, volume 500-249 of *NIST Special Publication*, pages 13–16, Gaithersburg, USA, nov. National Institute of Standards and Technology.