Multilayered Question Answering system applied to Temporality evaluation*

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

Resumen: Este artículo presenta una arquitectura multicapa de Búsqueda de Respuestas (BR) que permite mejorar los resultados de los sistemas actuales de BR ofreciendo la posibilidad de procesar preguntas complejas. Esto es, aquellas preguntas cuya respuesta se forma a partir de partes de información obtenidas de diferentes documentos. Concretamente, hemos diseñado una capa específica para procesar diferentes tipos de preguntas temporales. Las preguntas temporales complejas son en un primer paso descompuestas en preguntas simples, de acuerdo a las relaciones temporales expresadas en la pregunta original. De la misma manera, las respuestas de cada pregunta simple son recompuestas, en base a las restricciones temporales impuestas por la pregunta original compleja. Usando esta arquitectura, un sistema de BR Temporal ha sido desarrollado. En este artículo, nos hemos centrado en explicar la última parte de este proceso: la recomposición de las respuestas de las preguntas simples para conseguir la respuesta a la pregunta compleja. Además, esta unidad ha sido evaluada basándonos en la descomposición previa que hizo nuestro sistema del corpus de preguntas TERQAS formado por 124 preguntas temporales. Para obtener las respuestas a las preguntas descompuestas hemos utilizado un sistema de Búsqueda de Respuesta basado en web llamado IONAUT. Nuestro sistema mejora un sistema general contestando un 34 % de preguntas más que un sistema de BR genérico.

Palabras clave: Expresiones Temporales, Búsqueda de Respuestas, Anotación temporal

Abstract: This paper presents a multilayered Question Answering (QA) architecture suitable for enhancing current QA capabilities with the possibility of processing complex questions. That is, questions whose answer needs to be gathered from pieces of factual information that is scattered in different documents. Specifically, we have designed a specific layer oriented to process the different types of temporal questions. Complex temporal questions are first decomposed into simple questions, according to the temporal relations expressed in the original question. In the same way, the answers of each resulting simple question are recomposed, fulfilling the temporal restrictions of the original complex question. Using this architecture, a Temporal QA system has been developed. In this paper, we have been focused on explaining the last part of this process:the recomposition of the answers to the simple questions to obtain the final answer to the complex question. In addition, this unit has been evaluated using the previous decomposition that our system have made of TERQAS question corpus formed by 124 temporal questions. In order to obtain the answers to the simple questions we have used a general purpose QA system based on Web called IONAUT. Our system improves answering into a 34 % of questions compared to a General Purpose QA system.

Keywords: Temporal Expressions, Question Answering, Temporal Annotation

1. Introduction

Question Answering can be defined as the answering by computers to precise or arbitrary questions formulated by users. QA systems are especially useful to obtain a specific piece of information without the need to manually go through all the available documentation related to the searched topic. Research in Question Answering is mainly focused on the treatment of *factual questions* that require very specific items of data, such as dates, names of entities or quantities as answer ("What is the capital of Brazil?").

Temporal QA is not a trivial task due to the complexity temporal questions can achieve. Current operational QA systems can deal with simple factual temporal questions. That is, questions requiring a date as answer ("When did Bob Marley die?") or questions that involve temporal expressions in their formulation ("Who won the U.S. Open in 1999?"). Processing this kind of questions is usually accomplished by identifying explicit temporal expressions in questions and relevant documents in order to deal with temporal expressions that are necessary to answer the queries. Even though, it seems necessary to emphasize the system described in (Breck et al., 2000) as the only one that also uses implicit temporal expression recognition for QA purposes by applying the temporal tagger developed by Mani and Wilson (2000). However, questions referring to the temporal properties of the entities being questioned and the relative ordering of events mentioned in the questions are beyond the scope of current QA 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 QA system that achieves to answer complex temporal questions. This approach tries to imitate human's behavior when solving 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 the example complex question into two simple 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 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 final answer one of the answers for the first question (if there is any), whose associated date is included within the period of dates corresponding to the answer of the second question. That is, he would obtain the final answer by recomposing the respective answers to each simple question through the temporal signal in the original question (*during*).

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

The paper has been structured in the following way: first of all, section 2 presents our proposal of a taxonomy for temporal questions. Section 3 describes the general architecture of our temporal QA system. Section 4 deepens into the last part of the system, the recomposition unit which is thoroughly explained. Finally, the evaluation of the system 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 QA 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

[*] 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.

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 they are resolved by a QA System directly without pre or postprocessing of the question. 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 question, but there are one or more temporal expressions that need to be recognized, resolved and annotated. All that 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 establish temporal constraints in 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 simple 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. Multilayered Question-Answering System Architecture

Current QA system architecture does 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 multilayered architecture. This architecture increases the functionality of the current QA 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 possible to answer the original question.

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

- the 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 QA system,
- receiving the answers to the simple questions from the current General Purpose QA system,
- the filtering and comparison between sub-answers to build the final complex answer.

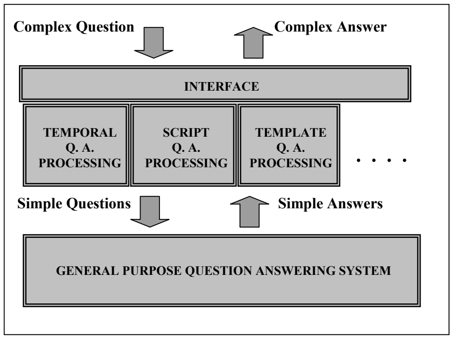

Figura 1: Multilayered Architecture of a QA

The main advantages of performing this multilayered system are:

- It allows you to use any existing general QA system, with the single effort of adapting the output of the processing layer to the type of input that the QA system uses.
- Due to the fact that the process of complex questions is done at a superior layer, is not necessary to modify the QA system when you want to deal with more complex questions.
- Each additional processing layer is independent from the others and it only processes those questions that are within the type of questions 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 (See Figure 2): Question Decomposition Unit, General purpose QA system and Answer Recomposition Unit.

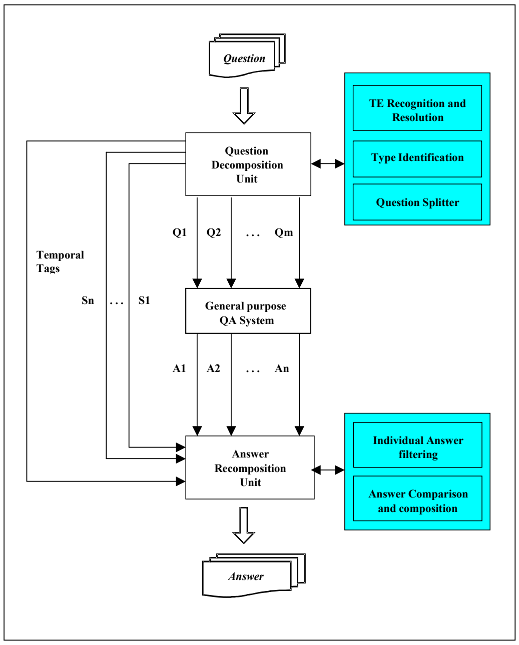
Question
TE Recognition and Resolution
Type Identification
Question Splitter
Question Decomposition Unit
Q1
Q2
. . .
Qm
Temporal Tags
Sn
. . .
S1
General purpose QA System
A1
A2
. . .
An
Answer Recomposition Unit
Individual Answer filtering
Answer Comparison and composition
Answer

Figura 2: Temporal Question Answering System

These components work all together in order to obtain a final answer. The Question Decomposition Unit and the Answer Recomposition Unit are the units that conform the Temporal QA layer which process the temporal questions, before and after using a General Purpose QA 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 is done. Secondly, regarding the taxonomy of the questions shown in section 2, there are different types of questions and every type has to be treated in a different way from the others. For this reason, type identification needs to be done. After that, complex questions, that are Type 3 and 4, are split into simple ones. These simple questions are 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 that are 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 QA system. Any QA system could be used here. We have used a general QA system that is available on the Internet: the IO Question Answering system[1]. The only condition is to know the kind of output returned by the system in order to adapt the layer interface. For the example above, a current QA 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 other temporal information obtained from the question, like temporal signals (that are explained later) or temporal expressions. Finally it manages to return the correct answer to the original question.

  Apart from proposing a taxonomy of temporal questions, we have presented a multilayered QA architecture suitable for enhancing current QA capabilities with the possibility of adding new layers for processing different kinds of complex questions. Moreover, we have presented a specific layer oriented to process the different types of temporal questions.

  The last aim of this paper is to present in depth and evaluate the last part of the temporal question processing layer: the Answer Recomposition Unit.

In next section, the different parts of the unit are shown as well as some examples of their behavior.

4. Answer Recomposition Unit

The recomposition of the answer is the last stage in our system. The complex questions were divided by the Decomposition Unit into simple ones with successful results of precision and recall. These simple questions were processed by a QA system which returns a set of answer and passages of documents where the answer is contained. The Answer Recomposition Unit needs a preprocessing of this amount of information in order to relate the answers with a date. This date is obtained from the set of passages related with the answer because this date is necessary in order to filter the answers by the Individual Answer Filtering unit. In this unit, the temporal constraints imposed by the Temporal Expressions of the question are applied to all the answer and some wrong answers are rejected. Finally, using the ordering key imposed by the Temporal Signal of the complex questions, the single answers are ordered and a final answer to the complex question is obtained. This process is divided in three modules:

- Preprocessing of the QA system output
- Individual Answer Filtering, and
- Answer Comparison and Composition

This unit has as input the set of individual answers and the temporal tags and signals related with the question, information that is needed to obtain the final answer.

4.1. Preprocessing of the QA system output

Because our system is independent of the General Purpose QA system used to answer the questions, a preprocessing module will be necessary in order to format these answers to the specific structure that the recomposition unit is waiting for. The kind of input that the recomposition unit is waiting is a file with all the possible answers to the questions and the dates related to these answers. To obtain these dates, TERSEO system has been applied to the document passages where the answer is found. TERSEO system is used to recognize, annotate and resolve all the temporal expressions in the passages so that it is possible to obtain a date of occurrence of the event the system is asking about. The system looks for the event in the document and obtains the date related to this event. Once the answer and the date of the answer are obtained, the recomposition can be done.

[1] http://www.ionaut.com:8400/

4.2. Individual Answer Filtering

All the possible answers given by the General Purpose QA system are the input of the Individual Answer Filtering. For the sub-questions with a temporal expression, it selects only those answers that satisfy the temporal constraints obtained by the T.E. Recognition and Resolution Unit as temporal tags. The date of the answer should be between the VALDATE1 and VALDATE2 values of the tag to be selected by the module. Only those answers that fulfill the constraints go to the Answer Comparison and Composition module.

4.2.1. Example

In the example: *Where did Bill Clinton study before going to Oxford University in the sixties?*:

1. The first step of the Temporal QA system (Decomposition Unit) divided the question into two simple questions. The first one, Q1 is *"Where did Bill Clinton study?"* and the second one, Q2 is *"When did Bill Clinton go to Oxford University in the sixties?"*.
2. Some answers for each sub-question are obtained from the General Purpose Question-Answering system. After the preprocessing of these results, every answer is related to a date.
   - Answer for Question 1: Georgetown University(1964-1968)
   - Answer for Question 1: Oxford University (1968-1970)
   - Answer for Question 1: Yale Law School (1970-1973)
   - Answer for Question 2: 1968 (1968)
3. These answers are filtered by this unit, because they have to fulfill the temporal constraints. Temporal constraints were obtained by the Decomposition Unit. In this example, the temporal expression: *"in the sixties"* was recognized, annotated and resolved establishing that the date of a possible answer has to be between *VALDATE1:"01/01/1960"* and *VALDATE2: "12/31/1960"*. Only the first two answers of the Q1 fulfill the temporal constraint so, they are used to compose the final answer. The answer of the second question is inside the possible range of dates, so, it will be used for the recomposition of the final answer as well.

4.3. Answer Comparison and Composition

Finally, once the answers are filtered, using the signals and the ordering key implied by these signals, the results for every sub-question are compared by the Comparison and Composition Unit. Temporal signals denote the relationship between the dates of the events that they relate. 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 a certain order between these events, which is called *ordering key*. An example of some ordering keys are shown in Table 1.

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

Cuadro 1: Example of signals and ordering keys

Only the answers that fulfill the order established by the ordering key can be a possible answer for the original questions. These selected answers are composed. Due to that, the system would have been able to solve those complex temporal questions that were not treated at first.

4.3.1. Example

Continuing with the same example: *Where did Bill Clinton study before going to Oxford University in the sixties?*:

1. The answers are ordered using the ordering key returned by the temporal signal *"before"*. This signal means that the date of the answer of the first question is previous to the date of the answer of the second question.
2. After doing that, the final answer is obtained, in this case: *"Georgetown University"*, because the date of the answer of this first question is the only one previous to the date related to the sec-

ond question,which interval of dates was from *"01/01/1968"* to *"12/31/1970"*.

4.4. Recomposition Unit Evaluation

In this section, an evaluation of the Answer Recomposition Unit, that uses the answer of split complex questions to obtain a final answer is shown. For this evaluation, we needed a corpus of questions that contains as much simple questions as complex questions. Because question corpora used in TREC (TREC,) and CLEF (CLEF,) do not contain complex questions, we chose the TERQAS question corpus (Radev y Sundheim, 2002), (Pustejovsky, 2002) that consists of 124 complex temporal questions. This set of questions was split by the Question Decomposition Unit into simple questions obtaining a 83 % of precision and recall. The answers to these simple questions were obtained by a General Purpose QA system[2] and they were recomposed by the Recomposition unit. The results can be classified in three main groups:

- The results are the same in both systems. That is because:
  - The QA system does not give back any answer for that question and therefore the TQA system does not give back anything either. There are 47 questions of that kind and the type of questions more affected are Type 1 and Type 2.
  - The TQA system returns the same answers as the QA system by itself. This it is exclusively the case of Type 1 questions since our system does not make any type of processing on those questions. There are 34 questions of this kind in our set.
- Our system obtains better results than the QA system. There are four different situations:
  - The TQA system does not give back any answer because, although QA system gives back a set of answers, none of them fulfill the temporal constraints imposed by the question and therefore none of these answer is the correct answer. This would be considered a success on our system. There are 12 questions of this kind, and 11 of them are questions of Type 2.
  - The QA system does not give back any answer and nevertheless, when splitting the question in simple ones and later reconstructing the answer, the TQA system is able to give an answer to the complex question. There is only 1 question of this kind.
  - The QA system returns wrong answers, nevertheless, when filtering the Temporal Expressions and splitting the question, more temporal information is obtained and the TQA system is able to answer properly to the complex question. There is only 1 question of this type.
  - The QA system returns a set of answers, but without considering the temporal information and the TQA is able to filter those answers and giving back just those that fulfill the temporal restrictions. Therefore, in this case, the TQA is answering better than the QA system.There are 28 questions of this type and they are questions of Type 2 and Type 4.
- Our system obtains worst results than the QA system
  - The QA system is able to answer but the TQA is not. That is because, when the complex question is divided into two simple ones, there are some keywords in the second questions that are not being used to ask for the first question and these keywords may be useful to find any answer. For example, in the question *"Who was Secretary of Defense during the Gulf War?"*, the system looks for the keywords: *"Secretary of Defense"* and *"Gulf War"* and returns answers like *"Colin Powell"*, *"George Bush"*, *"Dick Cheney"*. But, using the TQA system, the question is divided into two simple questions: *"Who was Secretary of Defense?"* and *"When did the Gulf Was occur?"*. When the first question is processed the results are

[2] http://www.ionaut.com:8400/

	TOTAL	BETTER RESULTS	EQUAL RESULTS	WORST RESULTS	BETTER RESULTS (%)	WORST RESULTS (%)
Type 1 Questions	47	-	47	-	-	-
Type 2 Questions	59	36	23	-	61 %	-
Type 3 Questions	3	-	3	-	-	-
Type 4 Questions	15	6	8	1	40 %	6 %
All Questions	124	42	81	1	34 %	0.8 %

Cuadro 2: Evaluation of the system

not good because the information given by the keyword *"Gulf War"* is missed.Anyway, there is only 1 question with this problem.

The results of this study are shown in Table 2. As a conclusion, it could be said that our system is improving a General Purpose QA system in the 34 % of the questions and it works worst only in less that 1 % of the questions.

5. Conclusions

This paper presents a new and intuitive method for answering complex temporal questions using an embedded current factual-based QA system. This method is based on a new proposal 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 multilayered architecture that enables to solve complex questions by enhancing current QA capabilities.

The multilayered approach can be applied to any kind of complex questions that allow question decomposition such as script questions ("How do I assemble a bicycle?") or template-like questions ("Which are the main biographical data of Nelson Mandela?").

Specifically, the paper has been focussed on the recomposition of the answers of the simple questions, that were divided by the Decomposition unit, in order to obtain an answer for the complex questions that a General Purpose QA system is not able to resolve.

In future, our work is directed to fine tune this system and increase system capabilities in order to be able to process more kinds of complex questions.

Bibliografía

Breck, E., J. Burger, L. Ferro, W. Greiff, M. Light, I. Mani, y J. Rennie. 2000. Another sys called quanda. En *Ninth Text REtrieval Conference*, volumen 500-249 de *NIST Special Publication*, páginas 369–378, Gaithersburg, USA, nov. National Institute of Standards and Technology.

CLEF. Cross-language evaluation forum. `http://clef.iei.pi.cnr.it/`.

Mani, I. y G. Wilson. 2000. Robust temporal processing of news. En ACL, editor, *Proceedings of the 38th Meeting of the Association of Computational Linguistics (ACL 2000)*, Hong Kong, October.

Pustejovsky, J. 2002. Terqas:time and event recognition for question answering systems. `http://time2002.org/`.

Radev, D. y B. Sundheim. 2002. Using timeml in question answering. `http://www.cs.brandeis.edu/ j̃amesp/ arda/ time/ documentation/ TimeML-use-in-qa-v1.0.pdf`.

TREC. Text retrieval conference. `http://trec.nist.gov/`.