TALP-QA System at TREC 2004: Structural and Hierarchical Relaxing of Semantic Constraints

Daniel Ferrés, **Samir Kanaan**, **Edgar González**, **Alicia Ageno**, **Horacio Rodríguez**, **Mihai Surdeanu**, and **Jordi Turmo**

TALP Research Center
Universitat Politècnica de Catalunya
{*dferres,skanaan,egonzalez,ageno,horacio,surdeanu,turmo*}*@lsi.upc.edu*

Abstract

This paper describes TALP-QA, a multilingual open-domain Question Answering (QA) system under development at UPC for the past two years. The system is described and evaluated in the context of our participation in the TREC 2004 Main QA task. The TALP-QA system treats both factoid and definitional questions (other).

Factoid questions are resolved with a process consisting of three phases: question processing, passage retrieval and answer extraction. Our approach to solve this kind of questions is to build a semantic representation of the questions and the sentences in the retrieved passages. A set of semantic constraints are extracted for each question. The answer extraction algorithm extracts and ranks sentences that satisfy the semantic constraints of the question. If matches are not possible the algorithm relaxes the semantic constraints structurally (removing constraints) and/or hierarchically (abstracting the constraints using a taxonomy).

Definitional questions are treated in a three-stage process: passage retrieval, pattern scanning over the previous set of passages, and finally a filtering phase where only the most relevant and informative fragments are given as final output.

1 Introduction

This paper describes TALP-QA, a multilingual open-domain Question Answering (QA) system under development at UPC for the past 2 years. The paper focuses on our participation in the TREC 2004 evaluation. Our aim in developing TALP-QA has been to build a system as far as possible language independent, where language dependent modules could be substituted to allow the system to be applied to different languages. A first preliminary version of TALP-QA for English was used to participate in TREC 2003 QA track (see [Massot et al, 2003]). From this initial version, a new version for Spanish was built and was used in CLEF 2004 (see [Ferrés et al, 2004a]). An improved version, again for English, has been used in TREC 2004.

In this paper we present the overall architecture of TALP-QA and describe briefly its main components, focusing on those components that have been most changed since our initial prototype, and on those components that process English. We also present an evaluation of the system used in the TREC 2004 evaluation for both factoid and definitional questions.

2 System Description

2.1 Architecture Model

The system used for the TREC 2004 was partly running on the architecture proposed in [González, 2004]. The proposal of this architecture is to transform each component of the system into a server. Each server can be client to another system server providing a different service. The communication between clients and servers is managed by a central server called *MetaServer*.

Some of the advantages of this architecture are:

- It prevents from repeated initializations of the components.
- It allows easy debugging, maintenance and replacement of each individual component.

- Components only need to understand the communication protocol (which has been kept simple). This way, components of very different characteristics can be integrated into a single system in an easy way.
- Common components can be shared between different systems running on the same MetaServer.
- Eventually, this architecture can allow a distributed model of computing, where several machines with MetaServers communicate with each other. A Client could access a Server on a different machine in a completely transparent way.

This architecture has been designed to be application independent. It has been used, for instance, to integrate different modules of two multilingual summarization systems [Fuentes et al, 2004]. So far, only the first stages of the global QA process are running on this architecture. However, our intention is to eventually migrate all components of the QA system to it.

2.2 System Architecture

The system architecture follows the most commonly used schema, splitting the process into three phases that are performed sequentially. QA components may contain iterative algorithms (e.g. Passage Retrieval) but no feedback is propagated to the previous modules. There are three main subsystems (as shown in Figure 1), one corresponding to each phase:

1. Question Processing (QP)
2. Passage Retrieval (PR)
3. Answer Extraction (AE)

These subsystems are described below, but first we will describe some pre-processing tasks that were carried out on the document collection (the AQUAINT corpus) and the questions. As mentioned, our aim is to develop a language independent system. Language dependent components are included only in the Question Pre-processing and Passage Pre-processing components, and can be easily substituted by components for other languages.

2.3 Collection Pre-processing

We have used the *Lucene*[1] Information Retrieval (IR) engine to perform the PR task. Before TREC 2004 we indexed the whole AQUAINT collection (i.e. about 1 million documents). We pre-processed the whole collection with linguistic tools (described in sub-section 2.5) to mark the part-of-speech (POS) tags, lemmas and Named Entities (NE) of the text. This information was used to built an index with the following parts:

- Lemmatized text: this part is built using the word lemmas and the recognized Named Entities. This text is then indexed and used in the PR module.
- Original text: the original text with Named Entities that is retrieved when a query succeeds on the lemmatized text.

As an additional knowledge source that will be used in the AE task, an *idf* weight is computed at document level for the whole collection.

2.4 Target Analysis and Substitution

The original questions of the TREC 2004 QA track are guided by a target. Because our current QA system does not process questions within context, we designed a component to substitute all the references of the target in the original question with the target. A set of heuristics, implemented by means of regular expression patterns, has been applied to solve some forms of coreference. If the substitution is not possible, the target and its analysis (POS, lemma and NERC using the tools described in the next subsection) are added as a relevant information to the question analysis output.

2.5 Question Processing

The main goal of this subsystem is to detect the expected answer type and to generate the information needed for the other subsystems. For PR, the information needed is basically lexical (POS and lemmas) and syntactic, and for AE, lexical, syntactic and semantic. We use a language-independent formalism to represent all this information. In particular we use the same semantic primitives and relations for both languages (English and Spanish) processed by our system.

For TREC 2004 we used a set of general purpose tools produced by the UPC NLP group (see [Carreras et al, 2002], and [Carreras et al, 2004]) and another set of public NLP tools. The same tools are used for the processing of both the questions and the answer passages. The following components were used:

[1] http://jakarta.apache.org/lucene

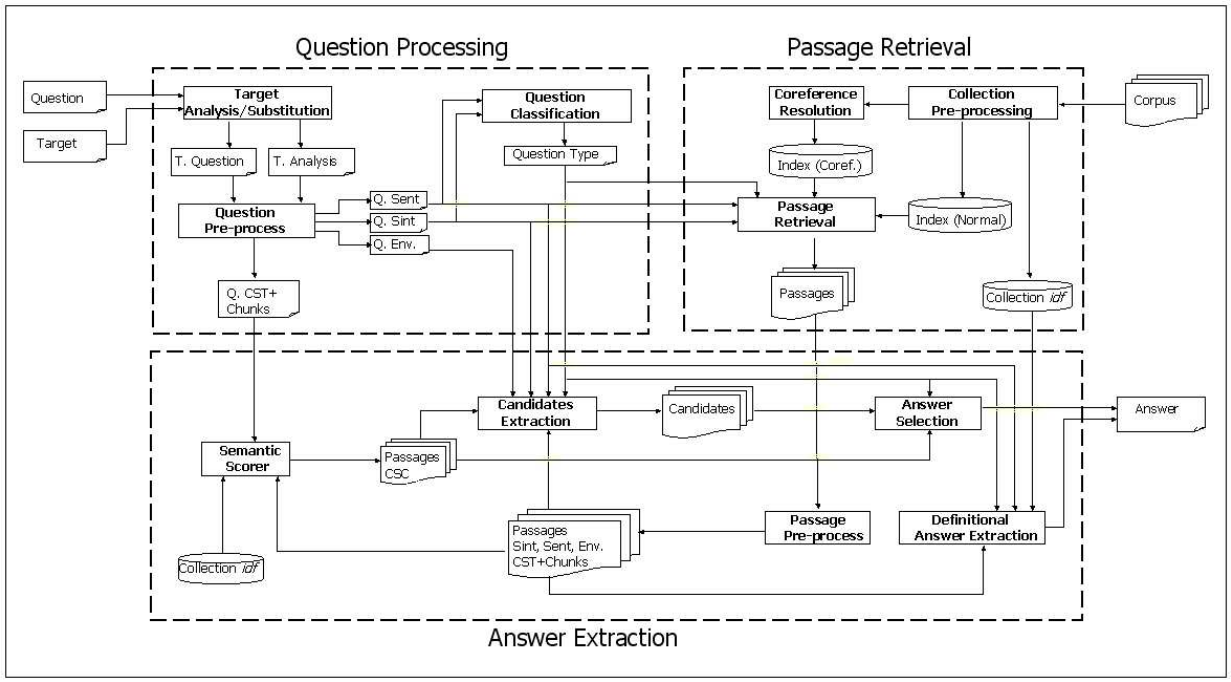

Figure 1: TALP-QA System Architecture.

- **Morphological components**, an statistical POS tagger (*TnT*) [Brants, 2000] and the WordNet lemmatizer (version 2.0) are used to obtain POS tags and lemmas. We used the *TnT* predefined model trained on the Wall Street Journal corpus.
- **A modified version of the Collins parser**, which performs full parsing and robust detection of verbal predicate arguments [Collins, 1999]. For the purpose of question answering, we have limited the number of predicate arguments to three: agent, direct object (or theme), and indirect object (benefactive or instrument), and use a series of robust heuristics to identify them. For example, one heuristic labels a noun phrase as agent if it precedes an active verb within a sentence construct. Furthermore, we have retrained the parser on a corpus of questions (SBARQ and SQ phrases) which are lacking in the original Penn TreeBank. From the previous TREC evaluations we have constructed an additional corpus of 1769 questions for training and a corpus of 537 questions for testing. Using this training corpus in addition to the TreeBank, our parser boosts its F-measure on the question test corpus from 81.82% to 95.10%.
- **ABIONET**, a Named Entity Recognizer and Classifier that identifies and classifies NEs in basic categories (person, place, organization and other). See [Carreras et al, 2002].
- **Alembic**, a Named Entity Recognizer and Classifier that identifies and classifies NEs with MUC classes (person, place, organization, date, time, percent and money). See [Aberdeen et al, 1995].
- **EuroWordNet**, used to obtain the following semantic information: a list of synsets (with no attempt to Word Sense Disambiguation), a list of hypernyms of each synset (up to the top of each hypernymy chain), the EWN's Top Concept Ontology (TCO) class [Rodríguez et al, 1998], and the Magnini's Domain Codes (DC) [Magnini, Cavagliá, 2000].
- **Three Gazetteers**, with the following information: acronyms, obtained using a Decision Tree approach [Ferrés et al, 2004b]; location-nationality relations (e.g. Spain-Spanish) and actor-action relations (e.g. write-writer).

The application of these linguistic resources and tools, obviously language dependent, to the text of the question (plus eventually to the target, either substituted for its coreferent or simply added to the analysis of the question itself) is represented in two structures (an example is presented in Figure 2):

- **Sent**, which provides lexical information for each word: form, lemma, POS, semantic class of NE, list of EWN synsets and, finally, whenever possible the verbs associated to the actor and the relations between locations and their nationality.
- **Sint**, composed by two lists, one recording the syntactic constituent structure of the question (including the specification of the head of each constituent) and the other collecting the information about relations between these components (in particular the subject, object and indirect object relations).

Once this information is obtained we can find the information relevant to the following tasks:

- **Question type**. The most important information we need to extract from the question text is the Question Type (QT), which is needed by the system when searching the answer. Failure to identity the QT practically disables the correct extraction of the answer. Currently we are working with about 26 QTs (we have used the same categories used in TREC 2003).

  The Question Types used were:

```
- abreviation
- abreviation_expansion
- definee
- definition
- event_related_to
- feature_of_person
- howlong_event
- howlong_object
- howmany_objects
- howmany_people
- howmuch_action
- non_human_actor_of_action
- subclass_of
- synonymous
- theme_of_event
- translation
- when_action
- when_begins
- when_person_died
- where_action
- where_location
- where_organization
- where_person_died
- where_quality
- who_action
- who_person_quality
```

  The QT focuses the type of expected answer and provides additional constraints. For instance, when the expected type of the answer is a person, two types of questions are considered, *Who_action*, which indicates that we are looking for a person who performs a certain action and *Who_person_quality*, that indicates that we are looking for a person having the desired quality. The action and the quality are the parameters of the corresponding QT. The following are examples of questions correctly classified respectively as *Who_person_quality* and *Who_action* type:

  - *Who was the head of the XII Israel government?*
  - *Who won the Nobel Prize for Literature in 1994?*

  In order to determine the QT our system uses an Inductive Logic Programming (ILP) learner that learns a set of weighted rules from a set of positive and negative examples. We used as learner the FOIL system [Quinlan, 1993]. A binary classifier (i.e. a set of rules) was learned for each QT. As training set we used the set of questions from TREC 8 and 9 (~900 questions) manually tagged and as test set the 500 questions from TREC 11. For each classifier we have used as negative examples the questions belonging to the other classes. For the classification task the following features were used: form, position in the question, lemma, POS, semantic class of NE, synsets together with all their hypernyms, TCO, DC and subject and object relations.

  The set of rules for each class was manually revised and completed with a set of manually built rules (with higher weights) in order to ensure a greater coverage. See below a couple of such rules:

  - A learned rule:

```
rule(non_human_actor_of_action,A,weight_1):-
  first_position(A,B),
  next_position(B,C),
  is_tco(cObject,C),
  is_domain(dTransport,C).
```

  - The same rule after transformation (performed for the sake of efficiency):

```
rule(non_human_actor_of_action,A,weight_1,
    [],TT) :-
  sent(A,_,TT), TT=[_,W2|_],
  has_tco(W2,cObject),
  has_domain(W2,dTransport).
```

    This rule executes as follows: using the question number (A), the *Sent* of the sentence is retrieved (TT). Then, the information about the second token from the

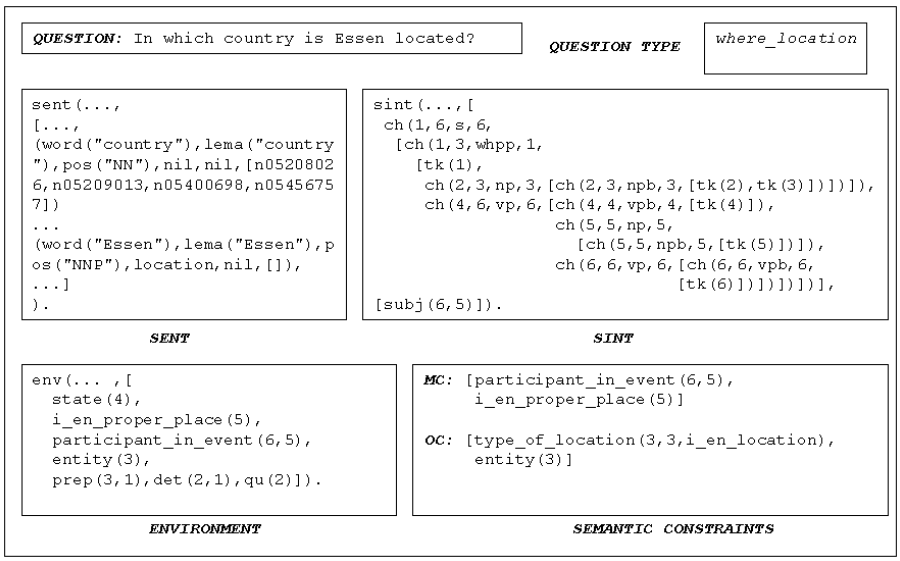

Figure 2: Results of the pre-process of a question.

sentence is obtained. Finally, we check that this token has a TCO corresponding to the class *Object* and a Domain Code corresponding to the class *Transport*.

– A manual rule:

```
rule(non_human_actor_of_action,A,weight_994,
    [T1,T3],T) :-
  sent(A,_,[T1|T]),
  the_lema(T1,lema("which")),
  has_chunk_with_hyperonym(_,T,[T2|TT],
    [sArtifact,sObject,sAnimal],T3),
  the_pos(T2,pos("IN")),
  not(has_term_with_pos(TT,pos("JJS"),_)).
```

The manual rule executes as follows: using the question number (A), the first token of the sentence ($T1$) and the following tokens of the sentence (T) are retrieved . Then, we check that the first token has "which" as lemma and the token's list T has a chunk with a token having an hypernym corresponding to one of the following synsets: artifact, object or animal. Finally, we check that the first token of the chunk ($T2$) is a preposition or a subordinating conjunction and does not contain a superlative adjective in its text.

- **Environment.** The semantic process starts with the extraction of the semantic relations that hold between the different components identified in the question text, eventually enriched with the target, either substituted or added to the question *sent* as mentioned before. These relations are organized into an ontology of about 100 semantic classes and 25 relations (mostly binary) between them. Both classes and relations are related by taxonomic links. The ontology tries to reflect what is needed for an appropriate representation of the semantic environment of the question (and the expected answer). For instance, *Action* is a class and *Human_action* is another class related to *Action* by means of an *is_a* relation. In the same way, *Human* is a subclass of *Entity*. *Actor_of_action* is a binary relation (between a *Human_action* and a *Human*). When a question is classified as *Who_action* an instance of the class *Human_action* has to be located in the question text and its referent is stored. Later, in the AE phase, an instance of *Human_action* co-referring with the one previously stored has to be located in the selected passages and an instance of *Human* related to it by means of the *Actor_of_action* relation must be extracted as a candidate to be the answer.

  The environment of the question is obtained from *Sint*, the semantic information included in *Sent* and EuroWordNet. A set of about 150 rules was built to perform this task. An example of environment extracted from a question is presented in Figure 2.

- **Semantic Constraints.** The Semantic Constraints Set (SCS) is the set of semantic relations that are supposed to be found in the sentences

containing the answer. The SCS of a question is built basically from its environment. The environment tries to represent the whole semantic content of the question while the SCS should represent a part of the semantic content of the sentence containing the answer. Mapping from the environment into the SCS is not straightforward. Some of the relations belonging to the environment are placed directly in the SCS, some are removed and some are modified (usually to become more general) and, finally, some new relations are added (e.g. *type_of_location*, *type_of_temporal_unit*,..., frequently derived from the question focus words). Relations of SCS are classified into two classes: Mandatory Constraints (MC) and Optional Constraints (OC). MC have to be satisfied in the answer extraction phase, OC are not obligatory, their satisfaction simply increases the score of the answer.

In order to build the semantic constraints for each question a set of rules (typically 1 or 2 for each type of question) has been manually built. The environment is basically a first order formula with variables denoted by natural numbers (corresponding to the tokens in the question). Several auxiliary predicates over this kind of formulas are provided and can be used in these rules. Usually these predicates allow the inclusion of filters, the possibility of recursive application and other generalization issues. A fragment of the rule applied in the example is presented in Figure 3. The rule can be paraphrased as follows: If the relation *state(C)* holds in the environment, get recursively all the predicates related to C, then filter the appropriate ones to be included in MC and OC and finally extend these sets for the sake of completeness. The application of the rule results in the constraints shown in Figure 2. The binary and unary predicates that compose the environment are shown in this Figure. The unary predicates extracted are:

- *state(4)*: which corresponds to the verb "is".
- *i_en_proper_place(5)*: which corresponds to the Named Entity "Essen". This unary predicate specifies that Essen is a NE classified as a location.
- *entity(3)*: a common noun corresponding to "country".
- *qu(2)*: corresponds to the interrogative pronoun "which".

The binary predicates extracted are:

- *participant_in_event(6,5)*: a semantic relation between the Named Entity "Essen" and the verb "locate".
- *prep(3,1)* and *det(2,1)*: syntactic relations without semantic content.

From the binary and unary predicates the SCS extracted is:

- *participant_in_event(locate,Essen).* (MC)
- *i_en_proper_place(Essen).* (MC)
- *type_of_location(country,country, i_en_location).* (OC)
- *entity(country).* (OC)

2.6 Passage Retrieval

The main function of the passage retrieval component is to extract small text passages that are likely to contain the correct answer. Document retrieval is performed using the *Lucene* Information Retrieval system. For practical purposes we currently limit the number of documents retrieved for each query to 1000. The passage retrieval algorithm uses a data-driven query relaxation technique: if too few passages are retrieved, the query is relaxed first by increasing the accepted keyword proximity and then by discarding the keywords with the lowest priority. The reverse happens when too many passages are extracted. Each keyword is assigned a priority using a series of heuristics fairly similar to [Moldovan et al, 1999]. For example, a proper noun is assigned a priority higher than a common noun, the question focus word (e.g. "state" in the question "What state has the most Indians?") is assigned the lowest priority, and stop words are removed.

2.7 Factual Answer Extraction

After PR, for factual AE, two tasks are performed in sequence: Candidate Extraction (CE) and Answer Selection (AS). In the first component, all the candidate answers are extracted from the highest scoring sentences of the selected passages. In the second component the best answer is chosen.

- **Candidate Extraction**. The process of extraction of the answer is presented in Figure 4. The process is carried out on the set of passages obtained from the previous subsystem. First,

```
mandatory(Q,ManIni,CRC,ManFi2,OptFi3,where_location,1) :-
  …
  state(C,Q,Env),
  …
  get_related_tokens_in_env_rec(C,Env,SS,[qu]),
  filter_tuple_tokens(SS,Manx1,_,Opt1,
          [theme_of_event,time_of_event,location_of_event,which_entity],
          []),
  …
  filter_related_tokens(SS,
          [
                    [human_participant_in_event(C,_X)],
                    [participant_in_event(C,_X), i_en_proper_person(_X)],
                    [participant_in_event(C,_X), i_en_proper_organization(_X)],
                    [participant_in_event(C,_X), i_en_proper_named_entity(_X)]
          ],
          Manrel),
  …
  extend_mandatory(ManFi,Opt,V4,V2,ManFi1,OptFi,Q,Env,Env),
  extend_mandatory_1(ManFi1,OptFi,V1,V2,ManFi2,OptFi2,V11,V22,Q,SS1,Env),
  …
```

Figure 3: Semantic constraints of a question.

these passages are segmented into sentences and each sentence is scored according to its semantic content using the $tf * idf$ weighting of the terms from the question and taxonomically related terms occurring in the sentence [Massot et al, 2003]. The linguistic process of extraction is similar to the process carried out on questions and leads to the construction of the environment of each candidate sentence.

Once the set of sentence candidates has been pre-processed the application of the extraction rules follows an iterative approach. In the first iteration all the Mandatory Constraints have to be satisfied by at least one of the candidate sentences. If the size of the set of candidate sentences satisfying the MC is smaller than a predefined threshold a relaxation process is performed and a new iteration follows otherwise the extraction process is carried out.

The relaxation process of the set of semantic constraint is performed by means of structural or semantic relaxation rules, using the semantic ontology. Two kinds of relaxation are considered: i) moving some constraint from MC to OC and ii) relaxing some constraint in MC substituting it for another more general in the taxonomy. Once the SCS is relaxed the score assigned to the sentences satisfying it is decreased accordingly.

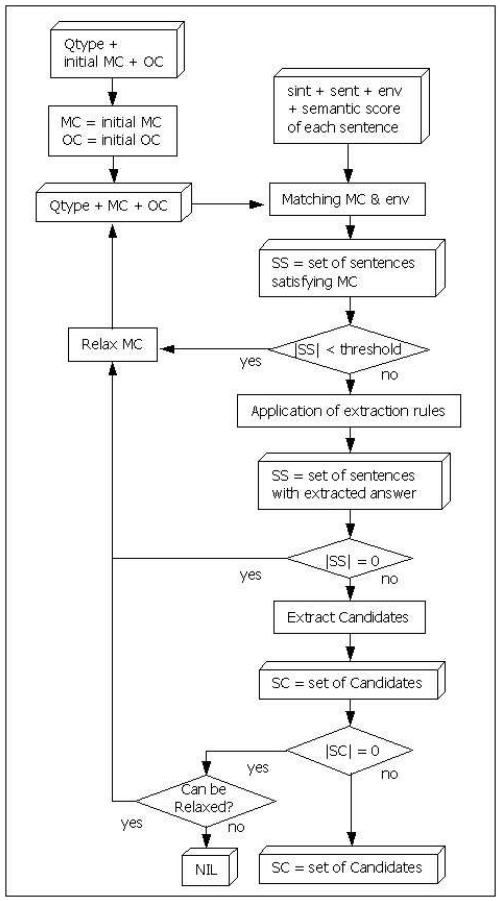

Figure 4: Candidates Extraction Relaxation Loop.

The extraction process consists on the application of a set of extraction rules on the set of sentences that have satisfied the MC. The Knowledge Source used for this process is a set of ex-

traction rules owning a credibility score. Each QT has its own subset of extraction rules that leads to the selection of the answer. An example of extraction rule is presented in Figure 5. The rule can be paraphrased as follows: Look in MC for predicates *state(C)* and *location(X)* satisfied in the environment. Then look in the environment for the predicates, related to C, *location_of_event* and *location*. Assure that the two locations are different and adjust the corresponding score.

If no answer is extracted from any of the candidates a new relaxation step is carried out followed by a new iteration step.

If no sentence has satisfied the MC or if no extraction rule succeeds when all possible relaxations have been performed the question is assumed to have no answer.

- **Answer selection.** In order to select the answer from the set of candidates, the following scores are computed for each candidate sentence:
  - The rule score, which uses factors such as the confidence of the rule used, the relevance of the OC satisfied in the matching, and the similarity between NEs occurring in the candidate sentence and the question.
  - The passage score, which uses the relevance of the passage containing the candidate.
  - The semantic score, defined previously.
  - The relaxation score, which takes into account the level of rule relaxation in which the candidate has been extracted.

  For each candidate the values of these scores are normalized and accumulated in a global score. The answer to the question is the candidate with the best global score.

2.8 Definitional QA

The structure of our definitional QA subsystem participating in TREC 2004 resembles a pipeline consisting of three main phases:

1. **Passage Retrieval**: most relevant passages with respect to the question target are retrieved. Resolution of coreference is also included in our second run, in order to capture indirect references to the question target and thus maximize the number of relevant passages retrieved. Refer to section 2.6 to see a description of this phase.
2. **Pattern Detection**: the passages from the previous phase are scanned through in search of fragments likely to give important information about the target. Fragments that match some given patterns are selected in this phase.
3. **Fragment Filtering**: a further selection is made on the fragments from phase 2. The passages selected are expected to be the most informative and redundant passages are also excluded.

2.8.1 Coreference Resolution

In order to test the influence of Coreference Resolution in our system, one of the runs we sent took Coreference into account in the Passage Retrieval stage. At a previous stage a Coreference Resolution algorithm ran on all the 3rd person pronouns of the AQUAINT corpus. This algorithm worked in 3 steps:

1. A preprocessing phase, in which each document was tokenized, tagged and its NEs were recognized with the same tools we use for the normal preprocessing stage (section 2.5); and its Noun Phrases(NP) were detected using *Yamcha*[2].
2. A NE clustering phase, in which the NEs of the document were related using clustering information previously gathered [Ferrés et al, 2004b]. At the end of this process, a set of Referential Classes had been built, each one holding all the NEs considered to refer to the same entity. At the same time, information from the NE Classification[3] and a gazetteer of common first names[4] was used to determine the number, humanity and gender of each Referential Class.
3. A pronoun resolution phase, in which every 3rd person pronoun looked among the NPs of its sentence and the previous ones for an antecedent (cataphora was not considered). The candidates were filtered according to their number, humanity and gender, which had to match those of the pronoun. To chose the antecedent, the accepted ones were sorted using the criteria proposed by Mitkov [Mitkov, 1998]. If the score of the best ranked candidate was lower than a static threshold, the pronoun was considered to have no antecedent. Otherwise, it was added to its Referential Class.

[2] http://chasen.org/~taku/software/yamcha/
[3] PERSON, ORGANIZATION, LOCATION, OTHERS
[4] http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/corpora/names/

```
extract_contextual_answer_from_tokens(DS,SS,_,_,Env, where_location,1, MT,A1,Sc2,_):-
  satisfy_MT_esp_obl([state(C),location(X)],MT,_),Sc=10,
  satisfy_strict([location_of_event(C,A,DS,Env),location(A,DS,Env)]),
  X\==A,
  nth(A,SS,A1),
  nth(X,SS,A2),
  smooth_scr(SS,X,A,Sc,Sc1),
if(
satisfy_MT_esp_obl([type_of_location(_,_,TL)],MT,_),
(check_type_of_location(A1,TL,A2,Sc3),Sc3 > 0.4, Sc2 is (Sc1 + Sc3 * 10)/ 2),
Sc2 is Sc1).
```

Figure 5: One of the extraction rules applicable to the example.

The final step was reindexing the AQUAINT corpus using as lemma for the resolved pronouns the representative of its Referential Class. However, as our goal was to recover extra passages that could contain information about a given target, we wanted a system with high precision, even if it was at the cost of recall. After experimentation, it was decided that the only pronouns to be substituted were the 3rd person singular ones (he, she, it), and only when its referent was a NE (not another kind of NP).

2.8.2 Pattern Detection

The objective of this phase is to identify candidate fragments by their concordance to a definitional pattern. These patterns have been elaborated manually, although by means of an automatically assisted procedure. Successful definitional QA solutions have already been developed, such as [Xu et al, 2003]. The passages retrieved are compared against the definitional patterns, which range from very general patterns such as those aimed at the detection of appositions, to specific patterns expecting exact verb matches.

2.8.3 Fragment Filtering

A further filtering of fragments is necessary in order to improve the precision of selecting the most informative fragments and excluding redundant ones. First, fragments are ordered by their relevance with respect to the question target. A list of words more closely related to the question target is created, and then fragments are ranked by the presence of these keywords. Second, highest-ranked fragments are selected sequentially and redundant ones are excluded. The redundance is interpreted as the overlap greater than 70% in non-stop words. This method expects to reduce redundant fragments, keeping the most relevant as a result.

3 Results

This section evaluates the behaviour of our system in TREC 2004. We evaluated the three main components of our system and the global results:

- **Question Processing**. This subsystem has been manually evaluated for factoid questions (see Table 1) and the following components: target analysis (NERC) and substitution in the original question, basic NLP tools (POS, NER and NEC), semantic pre-processing (Environment, MC and OC construction) and finally, question classification.

  In the following components the errors are cumulative: basic NLP tools (NE Recognition is influenced by POS-tagging errors and NE Classification is influenced by NE Recognition and POS-tagging errors), semantic pre-processing (the construction of the environment depends on the errors in the basic NLP tools and the syntactic analysis, the MC and OC errors are influenced by the errors in the environment), and question classification (is influenced by the errors in the basic NLP tools and the syntactic analysis).

Subsystem	Accuracy
Target Substitution	91.52% (151/165)
Target Analysis	72.31% (47/65)
POS-tagging	97.89% (1621/1656)
NE Recognition	88.89% (184/207)
NE Classification	82.13% (170/207)
Environment	45.22% (104/230)
MC	41.74% (96/230)
OC	82.61% (190/230)
Q. Classification	74.34% (171/230)

Table 1: Results of Question Processing evaluation.

- **Passage Retrieval**. The evaluation of this subsystem was performed using the set of correct an-

swers given by the TREC organization (see Table 2). We submitted two runs. In the first run (run1) we used the normal index in the Passage Retrieval phase for factoid and 'other' questions. In the second run (run2) we used the normal index for factoid questions and the Coreference index for 'other' questions. In both runs we retrieved only the 50 top passages for factoid and 'other' questions. These passages were selected from the 1000 top documents.

Accuracy Measure	Result
Factoid (*answer*)	72.41% (147/203)
Factoid (*answer+docID*)	58.62% (119/203)
Other (*vitals*) run1	58.55% (137/234)
Other (*okays*) run1	49.13% (170/346)
Other (*total nuggets*) run1	52.93% (307/580)
Other (*vitals*) run2	60.68% (142/234)
Other (*okays*) run2	49.13% (170/346)
Other (*total nuggets*) run2	53.79% (312/580)

Table 2: Results of Passage Retrieval for Factoid and Other questions.

We designed a set of different measures to evaluate the Passage Retrieval for the following question types:

- Factoid Questions. In this part we computed two measures: the first one (called *answer*) is the accuracy taking into account the questions that have a correct answer in its set of passages. The second one (called *answer+docID*) is the accuracy taking into account the questions that have a minimum of one passage with a correct answer and a correct document identifier in its set of passages.
- Other questions. The accuracy measure of the Passage Retrieval stage with respect to Other questions has been designed taking into account the needs and goals of this type of questions. Answers to Other questions must contain a set of vital nuggets and, optionally, some okay nuggets. Therefore, in order to evaluate the performance of the passage retrieval module, the number of nuggets contained in the passages retrieved has been counted, distinguishing by nugget type and also considering both nugget types globally. This measure gives an indication of the maximum possible recall that can be achieved by the following stages of the Other questions subsystem.

- **Answer Extraction**. The evaluation of this subsystem for factoid questions has been done in three parts: evaluation of the Candidates Extraction (CE) module, evaluation of the Answer Selection (AS) module and finally we performed an evaluation of the AE subsystem's global accuracy for factoid questions in which the answer appears in our selected passages. The results are presented in Table 3.

Subsystem	Accuracy (*answer*)
Candidates Extraction	25.17% (37/147)
Answer Selection	83.78% (31/37)
Answer Extraction	21.08% (31/147)

Table 3: Factoid Answer Extraction results.

- **Global Results**. The overall results of our participation in TREC 2004 are listed in Table 4.

Measure	Results
Factoid Total	230
Factoid Right	36
Factoid Wrong	190
Factoid IneXact	4
Factoid Unsupported	0
Factoid Precision NIL	0.089 (5/56)
Factoid Recall NIL	0.227 (5/22)
Accuracy over Factoid	0.157
Average F-score List	0.031
Average F-score Other (Run1)	0.165
Average F-score Other (Run2)	0.197
Final score (run1)	0.128
Final score (run2)	0.136

Table 4: Results of TALP-QA system at TREC 2004.

4 Evaluation and Conclusions

This paper summarizes our participation in the TREC 2004 QA task. Our system obtained a final score of 0.128 in run1 and 0.136 in run2. We conclude with a summary of the system behaviour for the two question classes:

- **Factoid questions**. The accuracy over factoid questions is 15.7%. Although no direct compari-

son can be done with another evaluation on a different test set, we think that we have improved substantially our factoid QA system with respect to the results of the TREC 2003 QA task evaluation (5.3%).

  - **Question Processing**. The Question Classification subsystem has an accuracy of 74.34%. We improved slightly the results of this component with respect to the previous TREC evaluation. In the previous evaluation we obtained an accuracy of 69%.
  - **Passage Retrieval**. In the PR we evaluated that 72.41% of questions have a correct answer in their passages. The evaluation taking into account the document identifiers shows that 58.62% of the questions are definitively supported. The accuracy of our PR subsystem has improved because in the TREC 2003 evaluation we obtained an accuracies of 62.10% and 42.36% for the previous measures respectively.
  - **Answer Extraction**. The accuracy of the AE module for factoid questions for which the answer occurred in our selected passages is 21.08%. We achieved a significant improvement of our AE module, since the results of this component in TREC 2003 were 8.9%. We expect to improve these results by reducing the error rate in the construction of the *environment*, MC and OC.

- **Other questions**. The results for the questions in the 'other' category were 16.50% and 19.70% F-score in run1 and run2 respectively. The only difference between these runs was the index used in the Passage Retrieval phase. In run2 we used an index that had been pre-processed with a coreference resolution algorithm. A detailed analysis of the results of run2 is given below:

  Three patterns, expressing different kinds of appositions, correspond to 75.52% of the fragments selected and have produced 85.32% of the right fragments. Recall of the output of Passage Retrieval phase (58.55%) is reduced after pattern detection is applied to a 35.34%, due to the narrower selection of fragments. However, this also increases precision, from a 1.86% in PR phase to a 3.06% at the output of this phase, increasing F-score from 14.46% to 17.19%.

  Although fragment filtering implies the loss of some information nuggets (recall is reduced to 25.11%), it also increases precision sensibly (to 6.70%), thus achieving a final F-score of 19.70%, more than two points above the output of the previous phase.

Acknowledgments

This work has been partially supported by the European Commission (CHIL, IST-2004-506909) and the Spanish Research dept. (ALIADO, TIC2002-04447-C02). Our research group, TALP Research Center, is recognized as a Quality Research Group (2001 SGR 00254) by DURSI, the Research Department of the Catalan Government.

References

[Aberdeen et al, 1995] J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain.
MITRE: Description of the ALEMBIC System Used for MUC- 6.
In Proceedings of the 6th Message Understanding Conference, pages 141-155.
Columbia, Maryland, 1995.

[Bikel, 2004] D.M. Bikel.
Intricacies of Collins' Parsing Model.
Computational Linguistics, December 2004, vol. 30, no. 4, pp. 479-511(33).

[Brants, 2000] T. Brants.
A Statistical Part-of-Speech Tagger
Proceedings of the 6th ANLP-NAACL. Seattle, USA, 2000.

[Carreras et al, 2002] X. Carreras, L. Márquez and L. Padró.
Named Entity Extraction Using Adaboost.
Proceedings of the CoNLL-2002. Shared Task Contribution. Taipei, Taiwan. September 2002.

[Carreras et al, 2004] X. Carreras, I. Chao, L. Padró, and M. Padró.
FreeLing: An Open-Source Suite of Language Analyzers.
Proceedings of LREC-2004. Lisbon, Portugal, 2004.

[Collins, 1999] M. Collins.
Head-Driven Statistical Models for Natural Language Parsing.
PhD Dissertation. University of Pennsylvania, 1999.

[Ferrés et al, 2004a] D. Ferrés, S. Kanaan, A. Ageno, E. González, H. Rodríguez, M. Surdeanu, and J. Turmo.
TALP-QA System for Spanish at CLEF 2004: Structural and Hierarchical Relaxation over Semantic Constraints.
Results of the CLEF 2004 Evaluation Campaign, Springer-Verlag LNCS series, to appear.

[Ferrés et al, 2004b] D. Ferrés, M. Massot, M. Padró, H. Rodríguez, J. Turmo.
Automatic Building Gazetteers of Co-referring

Named Entities.
Proceedings of LREC-2004. Lisboa, Portugal, 2004.

[Fuentes et al, 2004] M. Fuentes, E. González, J. Turmo.
Baseline summarization system for text including speech transcripts.
European Project CHIL (IP 506909) Deliverable D5.8, 2005.

[González, 2004] E. González.
Un Sistema Genèric de Cerca de Resposta.
Degree Project in Computer Science. Universitat Politècnica de Catalunya, Barcelona, 2004.

[Magnini, Cavagliá, 2000] B. Magnini, G. Cavagliá
Integrating Subject Field Codes into WordNet.
Proceedings LREC-2000. Athens, Greece, 2000.

[Massot et al, 2003] M. Massot, D. Ferrés, H. Rodríguez.
QA UdG-UPC System at TREC-12.
Proceedings of the TREC 2003. Gaithersburg, Maryland, United States, 2003.

[Mitkov, 1998] R. Mitkov.
Robust pronoun resolution with limited knowledge.
Proceedings of the 36th conference on ACL. Montreal, Canada, 1998.

[Moldovan et al, 1999] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Gîrju, V. Rus.
LASSO: A Tool for Surfing the Answer Net.
Proceedings of the Text Retrieval Conference (TREC-8). Gaithersburg, Maryland, United States, 1999.

[Quinlan, 1993] J.R. Quinlan
FOIL: A midterm report.
Proceedings of the sixth European Conf. on Machine Learning. Springer-Verlag, 1993.

[Rodríguez et al, 1998] H. Rodríguez, S. Climent, P. Vossen, L. Bloksma, W. Peters, A. Alonge, F. Bertanga, A. Roventini.
The Top-Down Strategy for Building EuroWordNet: Vocabulary Coverage, Base Concepts and Top Ontology.
Computer and Humanities 32. 1998, Kluwer Academic Publishers.

[Xu et al, 2003] J. Xu, A. Licuanan, R. Weischedel.
TREC2003 QA at BBN: Answering Definitional Questions.
Proceedings of the TREC 2003. Gaithersburg, Maryland, United States, 2003.