TALP-UPC at TREC 2005: Experiments Using a Voting Scheme Among Three Hetereogeneous QA Systems

Daniel Ferrés*, **Samir Kanaan***, **David Dominguez-Sal**†, **Edgar González***, **Alicia Ageno***, **Maria Fuentes***, **Horacio Rodríguez***, **Mihai Surdeanu***, and **Jordi Turmo***

*TALP Research Center
Software Department
Universitat Politècnica de Catalunya
{dferres, skanaan, egonzalez}@lsi.upc.edu

† DAMA-UPC
Department of Computer Architecture
Universitat Politècnica de Catalunya
ddomings@ac.upc.edu

Abstract

This paper describes the experiments of the TALP-UPC group for factoid and 'other' (definitional) questions at TREC 2005 Main Question Answering (QA) task. Our current approach for factoid questions is based on a voting scheme among three QA systems: TALP-QA (our previous QA system), Sibyl (a new QA system developed at DAMA-UPC and TALP-UPC), and Aranea (a web-based data-driven approach). For defitional questions, we used two different systems: the TALP-QA Definitional system and LCSUM (a Summarization-based system).

Our results for factoid questions indicate that the voting strategy improves the accuracy from 7.5% to 17.1%. While these numbers are low (due to technical problems in the Answer Extraction phase of TALP-QA system) they indicate that voting is a succesful approach for performance boosting of QA systems. The answer to definitional questions is produced by selecting phrases using set of patterns associated with definitions. Its results are 17.2% of F-score in the best configuration of TALP-QA Definitional system.

1 Introduction

This paper describes the experiments of the TALP-UPC group for factoid and 'other' questions at TREC 2005 Main Question Answering (QA) task.

The current approach for factoid questions is based on a voting scheme among three QA systems: TALP-QA, Sibyl and Aranea. TALP-QA is a multilingual open-domain Question Answering system under development at UPC for the past three years (see [6] and [5]). The approach is based on the use of in-depth NLP tools and resources to create semantic information representation. Sibyl is a new QA system developed during the last year at DAMA-UPC and TALP-UPC. This system uses a set of robust NLP tools to exploit inherent discourse properties. Aranea [1] is a Web-based factoid QA system that uses a combination of data redundancy and database techniques [11].

For definitional questions we used two different approaches: the TALP-QA Definitional system and LCSUM. TALP-QA Definitional system is a three-stage process: passage retrieval, pattern scanning over the previous set of passages, and finally a filtering phase where redundant fragments are detected and excluded from the final output. LCSUM is a summarizer based on Lexical Chains (see [7]). We used this summarizer to extract relevant information about the targets.

We have not designed any system for list questions. List questions are processed as factoid questions, but selecting answers among the ranked candidates that have a score higher than a certain threshold.

Finally, we outline below the organization of the paper. In Section 2, we present the overall architecture of the different factoid QA systems used and the voting scheme used for this kind of questions. Then, the definitional systems used at TREC 2005 are presented in Section 3. In section 4 and 5, we present the experiments and results obtained by our official runs at TREC 2005. Finally, in Section 6 and 7 we describe our evaluation and conclusions about the systems and the experiments.

[1] Aranea is a QA system released under GPL. `http://www.umiacs.umd.edu/~jimmylin/downloads/`

2 Factoid QA Systems

In this section we describe our two factoid QA systems (TALP-QA and Sibyl), the Aranea QA System, and the voting scheme used. But first, we present the target substitution process.

2.1 Target Substitution

The original questions of the TREC 2005 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 to 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, then the target is added at the end of the question; following this pattern: Question + ”in the” + <TARGET> ?.

2.2 TALP-QA System

TALP-QA is a multilingual open-domain Question Answering (QA) system under development at UPC for the past three years (see [6] and [5]). The system architecture has three phases that are performed sequentially without feedback: Question Processing (QP), Passage Retrieval (PR) and Answer Extraction (AE).

The TALP-QA approach is based on in-depth NLP processing and semantic information representation. 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 or making them optional) and/or hierarchically (abstracting the constraints using a taxonomy).

The version used in TREC 2005 is almost identical to the version used in TREC 2004 [6], with the exception of the following modules: Question Classification, Document Indexing, and Answer Selection.

The main subsystems are described below, but first we will describe the processing tasks over the document collection and the questions.

2.2.1 Collection Pre-processing

We have used the *Lucene*[2] Information Retrieval (IR) engine to perform the PR task. We indexed the whole AQUAINT collection (i.e. about 1 million documents) and we computed the *idf* weight at document level for the whole collection.

We pre-processed the whole collection with linguistic tools (described in sub-section 2.2.2) to mark the part-of-speech (POS) tags, lemmas and Named Entities (NE) of the text. This information was used to build an index with two fields per document: i) the lemmatized text with POS tags, and the recognized Named Entities with its class, ii) the original text (forms) with Named Entity Recognition. The first field is used in a search by lemma, and the information of both fields is retrieved when a query succeeds.

2.2.2 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.

For TREC 2005 we used a set of general purpose tools produced by the UPC NLP group and another set of public NLP tools. The same tools are used for the processing of both the questions and the retrieved passages. The following components were used:

- **Morphological components**, an statistical POS tagger (*TnT*) [2] and the WordNet lemmatizer (version 2.0) are used to obtain POS tags and lemmas. We used the *TnT* pre-defined 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 (see [4]).

- **ABIONET**, a Named Entity Recognizer and Classifier that identifies and classifies NEs in basic categories (person, place, organization and other). See [3].

- **Alembic**, a Named Entity Recognizer and Classifier that identifies and classifies NEs with MUC classes (person, place, organization, date, time, percent and money). See [1].

- **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), and the EWN's Top Concept Ontology (TCO) class [14].

[2] `http://jakarta.apache.org/lucene`

- **Gazetteers**: 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 is represented in two structures:

- **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 among constituents (subject, object and indirect object relations).

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

- **Environment.** The semantic process starts with the extraction of the semantic relations that hold between the different components identified in the question text. 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).

  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.

- **Question Classification**. 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. The QT focuses the type of expected answer and provides additional constraints. Currently we are working with about 26 QTs.

  The Question Classification module is composed of 72 hand made rules. These rules use a set of introducers (e.g. 'where'), and the predicates extracted from the environment (e.g. location, state, action,...) to classify the questions.

- **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.

2.2.3 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 [13]. 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.

2.2.4 Factoid Answer Extraction

After PR, for factoid 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 is carried out on the set of passages obtained from the previous subsystem. First, 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 [12]. 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 preprocessed 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.

  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 extraction rules owning a credibility score. Each QT has its own subset of extraction rules that leads to the selection of the answer. 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** After all candidates have been extracted, a single one must be selected as the answer. For this process we have used Support Vector Machines ([16]).

  Specifically, we have used the framework for ranking defined in [8] and implemented in SVM-Light[3]. For each candidate we extract the following attributes: the relaxation level in which the candidate has been extracted, the rule which allowed the extraction of the candidate, the rule score, the semantic score, and the passage score. The ranking uses a linear kernel. The best ranked candidate is given as the answer.

  The SVM was trained with the corpora of questions from TREC8 to TREC12, a total of 2392 questions. These questions were processed with a version of our system and the obtained candidates were checked against the official answers provided in the TREC website. Of the 2392 questions, candidates were found for 1592, and only 222 questions had the right answer among their candidates.

2.3 Sibyl: Robust Harnessing of Discourse Properties

The Sibyl QA system implements a divergent approach from the TALP-QA system introduced in the previous section. While the first system described in this paper uses complex resources, e.g. full parsing and semantic dictionaries, to achieve an in-depth understanding of the answer and question texts, the second system we developed uses only robust NLP tools to exploit inherent discourse properties, such as locality and density of question keywords in the proximity of candidate answers.

This system follows the same framework previously introduced, i.e. Question Processing (QP) - Passage Retrieval (PR) - Answer Extraction (AE), but most of the components and the NLP resources employed are completely different. We chose not only to implement a radically different approach to QA but also to use different NLP tools, e.g. NERC, in order to maximize the differences between the individual QA systems, a key feature to successful voting. We describe the relevant components and resources used in the new system next.

[3] `http://svmlight.joachims.org`

2.3.1 Question Processing

The QP component implements two tasks: (i) it detects the type of the expected answer, and (ii) it converts the NLP question into a list of prioritized keywords to be used for document/passage retrieval.

The first task is implemented using a question classification framework largely inspired by [9]. Similarly to [9], we extract from each question to be classified a rich of set features as follows: (i) we generate unigrams and bigrams for all the question words; (ii) we generate unigrams and bigrams for the head words of all basic syntactic phrases detected in the question; (iii) in all the n-gram features created we identify the first word of the question and the question focus word, both of each give strong hints of the question type; and (iv) all lexicalized n-grams constructs are expanded using the semantic classes provided by [9] and the proximity-based thesaurus supplied by [10]. Unlike [9], we did not implement a hierarchy of classifiers, but rather opted for a single, flat classifier using Maximum Entropy. The classifier was trained using the training set of questions provided by [9], which includes 50 question classes. On the same testing data as [9], our classifier obtains an accuracy of 88.4%. Once a question is classified, the expected answer type is set using a mapping generated off-line from the 50 question classes to one of the 9 NE classes recognized by our NERC.

The selection of question keywords for passage retrieval is implemented using the heuristics for keyword priority reported by [13]. Essentially, we favor proper names over nouns, which in turn have a higher priority than verbs, etc. The lowest priority is assigned to the question focus word, which is unlikely to appear in candidate answers.

2.3.2 Passage Retrieval

The passage retrieval algorithm used by Sibyl is similar to the one used by TALP-QA. In a nutshell, we use an incremental query relaxation technique that adjusts both the keyword proximity and the number of keywords included in the query, until a certain number of documents and passages is retrieved. Because our answer ranking algorithm works better when more question keywords are found in the candidate answer contexts, we first relax keyword proximity and only if the desired number of documents/passages is not obtained with the largest acceptable proximity we discard lower priority keywords.

2.3.3 Answer Extraction

The answer extraction component ranks candidate answers based on the properties of the context where they appear in the retrieved passages. We consider as candidate answers all named entities of the same type as the answer type detected by the question processing component. Candidate answers are ranked using a set of six heuristics, inspired by [13]:

- *Same word sequence* - computes the number of words that are recognized in the same order in the answer context;
- *Punctuation flag* - true when the candidate answer is followed by a punctuation sign;
- *Comma words* - computes the number of question keywords that follow the candidate answer, when the later is succeeded by comma. The last two heuristics are a very basic detection mechanism for appositive constructs, a common form to answer a question;
- *Same sentence* - the number of question words in the same sentence as the candidate answer.
- *Matched keywords* - the number of question words found in the answer context.
- *Distance* - the largest distance (in words) between two question keywords in the given context. The last three heuristics quantify the proximity and density of the question words in the answer context, which are two intuitive measures of answer quality.

All these heuristics can be implemented without the need for any NLP resources outside of a basic tokenizer. For each candidate answer, these six values are then converted into an answer score using the formula proposed by [13].

2.3.4 NLP Resources

The two NLP tools required by this system are: recognition of basic syntactic phrases, i.e. chunking, for QP, and named entity recognition and classification (NERC) for AE.

The chunker used was trained using a series of one-versus-all classifiers for each syntactic category. Each classifier was implemented using Support Vector Machines (SVM) with a polynomial kernel of degree 2. We trained the chunker on the corpora provided by the 2000 CoNLL shared task [15]. On the testing data from the same evaluation exercise, our chunker obtains an F1 measure of 95.21.

The NERC employed by this QA system recognizes the following 9 semantic categories: location, person, organization, and other miscellaneous names; times and dates; monetary values; percents and numbers. The first four categories (all names) are recognized with a system very similar with the previously-introduced chunker: one-versus-all SVM classifiers trained on the CoNLL shared task data [15]. On the CoNLL testing data, this system obtains an F1 measure of 87.50. Temporal entities and percents are recognized with the Alembic system [1]. Finally, all other numbers are identified with an in-house system based on regular expression grammars.

2.4 Aranea

The QA system Aranea took part in TREC 2002, TREC 2003 and TREC 2004 evaluations, and is described in [11]. Aranea uses two different techniques:

- **Knowledge Annotation**. For some very frequent fixed-pattern questions such as *What is the population of X?*, *What is the atomic symbol of X?* or *Who was the second president of the United States?*, structured databases available in the web, such as the CIA World Factbook or `biography.com` are queried. The question is transformed into a query using simple patterns.
- **Knowledge Mining**. For the rest of the questions, the keywords of the question are detected and a web search engine (`Google`, `Teoma`) is used to retrieve passages. As the web is extremely redundant, the answer is expected to be found expressed in terms similar to the question, and to be extractable using simple pattern matching techniques.

We executed the open source version of Aranea using `Google` and `Teoma` as search engines to get extra evidence for list and factoid question candidates. The answers provided by Aranea are not directly selectable, as they do not come from the AQUAINT corpus, yet can boost candidates coming from the other two QA systems.

2.5 Voting Scheme

Our voting algorithm selects the final answer to each factoid question from the lists of the best 20 candidates extracted by each one of our QA systems: TALP-QA and Sibyl. This is done in two separated stages: considering that the answer for a question is the pair $< c, d >$ of both the best candidate c answering the question and a document d in which it occurs, the first stage selects c. Given that this text can occur in a set of documents, the second stage selects document d as the most plausible one from the set.

In order to select the best candidate c, a score $s_1(c_i)$ is computed as follows for each different candidate c_i occurring in the lists of top 20 answers:

$$s_1(c_i) = \sum_{o \in Occ(c_i)} \frac{1}{ranking(o)}$$

where $Occ(c_i)$ is the set of occurrences o of candidate c_i in a) the list of top 20 candidates achieved by TALP-QA, b) the list of top 20 ones achieved by Sibyl, and c) the list of top 20 ones achieved by Aranea (used to take into account evidences of the answer in the web). Function $ranking(o)$ is the ranking in which occurrence o is located in these lists. So, this score promotes those candidates located highest in the lists of top 20. Taking into account these scores, the candidate with highest score is selected to be c.

Finally, in order to select the most plausible document d for the answer, a score $s_2(d_i)$ is computed for each document d_i in which the selected candidate c occurs. This scores is as follows:

$$s_2(d_i) = \sum_{o \in Occ(d_i, c)} \frac{1}{ranking(o)}$$

where $Occ(d_i, c)$ is the set of occurrences o of candidate c in document d_i. The document with highest score is selected to be d.

The same scores are used for list questions. However, for list questions a list of answers is provided. Firstly, the list of those candidates c_i extracted by systems TALP-QA and Sibyl achieving a score $s_1(c_i)$ higher than a threshold (80% was used) is selected. Then for each of these candidates, the best document d is selected, as explained for factoid questions.

3 Definitional QA Systems

We describe below our two approaches for definitional QA: TALP-QA Definitional and LCSUM.

3.1 TALP-QA Definitional System

The TALP-QA Definitional system has three steps: first, the 50 most relevant documents with respect to the target are retrieved, from which the passages referring to the target are retrieved; second, sentences referring to the target are extracted from the previous set of documents, and last, redundant senteces are removed from the final output of the system.

3.1.1 Document and Passage Retrieval

An index of documents has been created using Lucene that searches using lemmas instead of words. The search index has two fields: one with the lemmas of all non-stop words in the documents, and another with the lemmas of all the words of the documents that begin with a capital letter. The target to define is lemmatized, stopwords are removed and the remaining lemmas are used to search into the index of documents. Moreover, the words of the target that begin with a capital letter are lemmatized; the final query sent to Lucene is a complex one, composed of one subquery using document lemmas and another query containing only the lemmas of the words that begin with a capital letter. This second query is intended to search correctly the targets that, although being proper names, are composed or contain common words. For example, if the target is ’Liberty Bell 7’, documents containing the words ’liberty’ or ’bell’ as common names are not of interest; the occurrence of these words is only of interest if they are proper names, and as a simplification this is substituted by the case the words begin with a capital letter. The score of a document is the score given by Lucene. Once selected a number of documents (50 in the current configuration), the passages (blocks of 200 words) that refer to the target are selected for the next phase.

3.1.2 Sentence Extraction

The objective of the second phase is to obtain a set of candidate sentences that might contain interesting information about the target. As definitions usually have a certain structure, as appositions or copulative sentences, a set of patterns has been manually developed in order to detect these and other expressions usually associated with definitions (for example, ’<phrase> , <target>’, or ’<phrase> be <target>’). The sentences that match any of these patterns are extracted.

3.1.3 Sentence Selection

In order to improve precision in the system’s response, redundant sentences are removed from the set of extracted sentences from the previous step. The redundance detection first creates a set with the first sentence (a sentence from the best scored document) and then adds to the set all the sentences whose word coincidence with the sentences in the set does not exceed a certain threshold.

3.2 LCSUM System

LCSUM is a summarizer based on Lexical Chains (see [7]). We used the English version of this system to extract relevant information about the targets. The summarization system receives as input the passages extracted by the Passage Retrieval module of the TALP-QA system. Firstly, for each target, the summarizer uses all the passages extracted in all the questions related to the target. Then, the lexical chains are computed for each passage related to the target. Finally, the first sentence with some word in a lexical chain is selected as a summary of the passage, trying not to exceed 300 words.

4 Experiments

We designed a set of experiments for factoid, list, and ’other’ questions (see Table 1). Concretely, we submitted 3 runs: *run1* (talpupc05a), *run2* (talpupc05b) and *run3* (talpupc05c).

The first run (*run1*) uses the TALP-QA system for factoid and list questions, and the TALP-QA Definitional system for ’other’ questions. The second run (*run2*) consists of a voting scheme among TALP-QA and Sibyl for factoid and list QA, and another configuration for the TALP-QA Definitional system for ’other’ questions. Finally, the third run (*run3*) uses a voting scheme among TALP-QA, Sibyl and Aranea for factoid and list QA, and the LCSUM summarizer for ’other’ questions.

The Document Ranking experiments were the following: *run1* uses only documents from the TALP-QA Factoid and Definitional systems. The second run (*run2*) uses documents from both TALP-QA and Sibyl for factoid and list, and TALP-QA Definitional for others. The third run (*run3*) uses documents from both TALP-QA and Sibyl for factoid and list (because Aranea do not provide documents from AQUAINT) and TALP-QA for ’other’ (because LCSUM uses passages retrieved by TALP-QA system). Finally, due to the fact that the task was not required to retrieve documents from ’other’ questions, *run2* and *run3* were identical.

Run	Factoid QA	Other QA
run1	TALP-QA	TALP-QA Def. (1)
run2	TALP-QA & Sibyl	TALP-QA Def. (2)
run3	TALP-QA & Sibyl &Aranea	LCSUM

Table 1: Experiments at TREC 2005 QA Main Task.

5 Results

This section presents the evaluation of the TALP-QA system for factoid Questions and the global results at TREC 2005.

- **Question Processing**. This subsystem has been manually evaluated for factoid questions (see Table 2) and the following components: target substitution in the original question, basic NLP tools (POS, NER and NEC), semantic preprocessing (Environment, MC and OC construction) and finally, Question Classification (QC).

  In the following components the errors are cumulative: basic NLP tools (NER is influenced by POS-tagging errors and NEC is influenced by NER and POS-tagging errors), semantic preprocessing (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 QC (is influenced by the errors in the basic NLP tools and the syntactic analysis).

Subsystem	Accuracy
Target Substitution	89.83% (309/344)
POS-tagging	98.87% (3149/3185)
NE Recognition	93.53% (434/464)
NE Classification	82.11% (381/464)
Environment	49.45% (179/362)
MC	31.77% (115/362)
OC	58.01% (210/362)
Q. Classification	76.79% (278/362)

Table 2: Results of Question Processing evaluation for the TALP-QA system.

- **Passage Retrieval**. The evaluation of this subsystem was performed using the set of correct answers given by the TREC organization (see Table 3).

Question	Accuracy	Result
Factoid	(*answer*)	62.60% (216/345)
(run1)	(*answer+docID*)	46.37% (160/345)

Table 3: TALP-QA Passage Retrieval results.

  We designed two different measures to evaluate the Passage Retrieval for Factoid questions: 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.

- **Answer Extraction**. We evaluated the Candidates Extraction (CE) module, 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.

Subsystem	Accuracy (*answer*)
Candidates Extraction	8.11% (28/345)
Answer Selection	71.42% (20/28)
Answer Extraction	5.79% (20/345)

Table 4: TALP-QA Answer Extraction results.

- **Global Results**. The overall results of our participation in the TREC 2005 Main QA Task are listed in Table 5. The results of Document Ranking Evaluation Task are listed in Table 6.

Measure	run1	run2	run3
Factoid Total	362	362	362
Factoid Right	27	53	62
Factoid Wrong	330	288	279
Factoid IneXact/Uns.	4/1	17/4	17/4
Factoid Precision NIL	7/172	5/76	5/77
Factoid Recall NIL	7/17	5/17	5/17
Accuracy over Factoid	0.075	0.146	0.171
Average F-score List	0.024	0.026	0.028
Average F-score Other	0.172	0.164	0.079
Final score	0.088	0.125	0.116

Table 5: Results of TALP's runs at TREC 2005.

Run	run1	run2
AvgP.	0.1191	0.1468
R-Prec.	0.1287	0.1685
Docs. Retrieved	781	1619
Recall (%)	11.68%	20%
Recall	184/1575	375/1575
Δ AvgP. Diff.(%) over all runs AvgP.	-32.15%	-7.22%

Table 6: TREC 2005 Document Ranking Task.

6 Evaluation

This section summarizes the evaluation of our participation in the TREC 2005 Main QA and Document Ranking tasks.

- **Question Answering Task**. Our system obtained a final score of 0.088 in *run1*, 0.125 in *run2*, and 0.116 in *run3* (see Table 5). We conclude with a summary of the system behaviour for each question class:

  - **Factoid Questions**. The accuracy over factoid questions is 7% in *run1*, 14.6% in *run2*, and 17.1% in *run3* (see 5). The results of the TALP-QA system (run1) are low due to errors in the Candidates Extraction module. Otherwise, the voting scheme is useful as seen in runs 2 and 3.

    The TALP-QA system (run1) has been evaluated in its three phases:

    i) **Question Processing**. The Question Classification subsystem has an accuracy of 76.79%. We improved slightly the results of this component with respect to the TREC 2004. In the previous evaluation we obtained an accuracy of 74.34%. These are good results if we take into account that in TREC 2005 has increased the average length of both questions and targets.

    ii) **Passage Retrieval**. We evaluated that 62.60% of questions have a correct answer in their passages. The evaluation taking into account the document identifiers shows that 46.37% of the questions are definitively supported. The accuracy of our PR subsystem has decreased in comparison with the TREC 2004 evaluation (72.41% and 58.62% of accuracy for the previous measures respectively). This drop may be due to the increase of the average question length at TREC 2005.

    iii) **Answer Extraction**. The accuracy of the AE module for factoid questions for which the answer occurred in our selected passages is 5.79%. This poor accuracy is due to a technical error in the AE module. Otherwise, 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 17.20%, 16.40%, and 7.9% F-score in *run1*, *run2* and *run3* respectively. The two runs with the TALP-QA Definitional system, had both similar results (17.2% and 16.4% of f-score), and they differ in the threshold applied in the sentence selection phase (70% and 60% respectively) in order to exclude redundant fragments from the final output of the system. LCSUM obtained a F-score of 7.9%, mainly because this summarizer has not its own Passage Retrieval system and used the passages retrieved by TALP-QA for factoid questions.

  - **List Questions**. The F-score over list questions is clearly poor : 2.4% in *run1*, 2.6% in *run2*, and 2.8% in *run3*.

- **Document Ranking Task**. The results of the Document Ranking task are presented in Table 6. Our system obtained an Average Precission of 0.1191 (run1) and 0.1468 (run2 and run3), a R-Precission of 0.1287 (*run1*) and 0.1685 (*run2* and *run3*). The Document Raking Median of over all runs of TREC 2005 was 0.1574. We obtained an Average Precission Difference over all runs of -32.15% (*run1*) and -7.22% (*run2* and *run3*).

7 Conclusions

We combined the results of three heterogeneous factoid QA Systems: TALP-QA (a precission-oriented QA system), Sibyl (a recall-oriented QA system) and ARANEA (a recall-oriented and Web-based QA system). The resulting voting scheme has been successful, improving the accuracy over *run1* with 108% in *run2* and with 144% in *run3*.

The results in factoid questions were 7% of accuracy in the run without voting, and 14.6% and 17.1% in the runs with voting. While these numbers are low (due to technical problems in the Answer Extraction phase of TALP-QA system) they indicate that voting is a succesful approach for performance boosting of QA systems.

Acknowledgments

This work has been partially supported by the European Commission (CHIL, IST-2004-506909), the Spanish Research Department (ALIADO, TIC2002-04447-C02), the Ministry of Universities, Research and Information Society (DURSI) of the Catalan Government, and the European Social Fund.

David Domínguez is granted by Catalan Government (2005FI 00437). Daniel Ferrés is supported by a UPC-Recerca grant from Universitat Politècnica de Catalunya (UPC). Mihai Surdeanu is a research fellow within the Ramón y Cajal program of the Spanish Ministry of Education and Science. Our research group, TALP Research Center, is recognized as a Quality Research Group (2001 SGR 00254) by DURSI.

References

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

[2] T. Brants. TnT – a statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference, ANLP-2000*, Seattle, WA, United States, 2000.

[3] Xavier Carreras, Lluís Màrquez, and Lluís Padró. Named Entity Extraction using AdaBoost. In *Proceedings of CoNLL-2002*, pages 167–170. Taipei, Taiwan, 2002.

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

[5] Daniel Ferrés, Samir Kanaan, Alicia Ageno, Edgar González, Horacio Rodríguez, Mihai Surdeanu, and Jordi Turmo. The TALP-QA System for Spanish at CLEF 2004: Structural and Hierarchical Relaxing of Semantic Constraints. In Carol Peters, Paul Clough, Julio Gonzalo, Gareth J. F. Jones, Michael Kluck, and Bernardo Magnini, editors, *CLEF*, volume 3491 of *Lecture Notes in Computer Science*, pages 557–568. Springer, 2004.

[6] Daniel Ferrés, Samir Kanaan, Edgar González, Alicia Ageno, Horacio Rodríguez, Mihai Surdeanu, and Jordi Turmo. TALP-QA System at TREC 2004: Structural and Hierarchical Relaxation Over Semantic Constraints. In *Proceedings of the Text Retrieval Conference (TREC-2004)*, 2005.

[7] Maria Fuentes and Horacio Rodríguez. Using cohesive properties of text for automatic summarization. In *Processings of the JOTRI-2002*, 2002.

[8] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, 2002.

[9] Xin Li and Dan Roth. Learning question classifiers: The role of semantic information. *Natural Language Engineering*, 1(1), June 2004.

[10] Dekang Lin. Proximity-based thesaurus. *http://www.cs.ualberta.ca/˜lindek/downloads.htm*, 2005.

[11] Jimmy Lin and Boris Katz. Question answering from the web using knowledge annotation and knowledge mining techniques. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 116–123, New York, NY, USA, 2003. ACM Press.

[12] Marc Massot, Horacio Rodríguez, and Daniel Ferrés. QA UdG-UPC System at TREC-12. In *Proceedings of the Text Retrieval Conference (TREC-2003)*, pages 762–771, 2003.

[13] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Goodrum, R. Gîrju, and V. Rus. LASSO: A tool for surfing the answer net. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 1999.

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

[15] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, 2000.

[16] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.