The University of Amsterdam at QA@CLEF 2003

Valentin Jijkoun Gilad Mishne Maarten de Rijke

Language & Inference Technology Group, University of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands
E-mail: {jijkoun, gilad, mdr}@science.uva.nl

Abstract

This paper describes the official runs of our team for QA@CLEF 2003. We took part in the monolingual Dutch Question Answering task.

1 Introduction

In this year's CLEF evaluation exercise we participated in the *Dutch Question Answering* task, new on the CLEF agenda, building on and extending our earlier work on question answering at TREC [6]. We experimented with a multi-stream architecture for question answering, in which the different independent streams, each a complete QA system in its own right, compete with each other to provide the system's final answer.

The paper is organized as follows. In Section 2 we describe the architecture of our system. Section 3 describes our official runs. In Section 4 we discuss the results we have obtained. Finally, in Section 5 we offer some preliminary conclusions regarding our Dutch question answering efforts.

2 System Description

The general architecture of a question answering (QA) system, shared by many systems, can be summed up as follows. A question is first associated with a *question type*, out of a predefined set such as DATE-OF-BIRTH or CURRENCY. Then a query is formulated based on the question, and an information retrieval engine is used to identify a list of documents that are likely to contain the answer. Those documents are sent to an *answer extraction* module, which identifies candidate answers, ranks them, and selects the final answer. On top of this basic architecture, numerous add-ons have been devised, ranging from logic-based methods [5] to ones that rely heavily on the redundancy of information available on the World Wide Web [2].

2.1 Multi-Stream Architecture

During the design of our QA system, it became evident that there are a number of distinct approaches for the task; some are beneficial for all question types, and others only for a subset. For instance, abbreviations are often found enclosed in brackets, following the multi-word string they abbreviate, as in "*Verenigde Naties (VN)*." This suggests that for abbreviation questions the text corpus can be mined to extract multi-word strings with leading capitals followed by capitalized strings in brackets; the results can then be stored in a table to be consulted when an abbreviation (or an expansion of an abbreviation) is being asked for. Similar table-creation strategies are applicable for questions that ask for capitals, dates-of-birth, etc., whereas the approach seems less appropriate for definition questions, why-questions, or how-to questions. It was therefore decided to implement a *multi-stream* system: a system that includes a number of separate and independent subsystems, each of which is a complete standalone QA system that produces ranked answers, but not necessarily for all types of questions; the system's answer is then taken from the combined pool of candidates.

Scientifically, it is interesting to understand the performance of each stream on specific question types and in general. On the practical side, our multi-stream architecture allows us to modify and test a stream without affecting the rest of the system. A general overview of our system is given in Figure 1. The system consists of 5 separate QA streams and a final answer selection module that combines the results of all streams and produces the final answers.

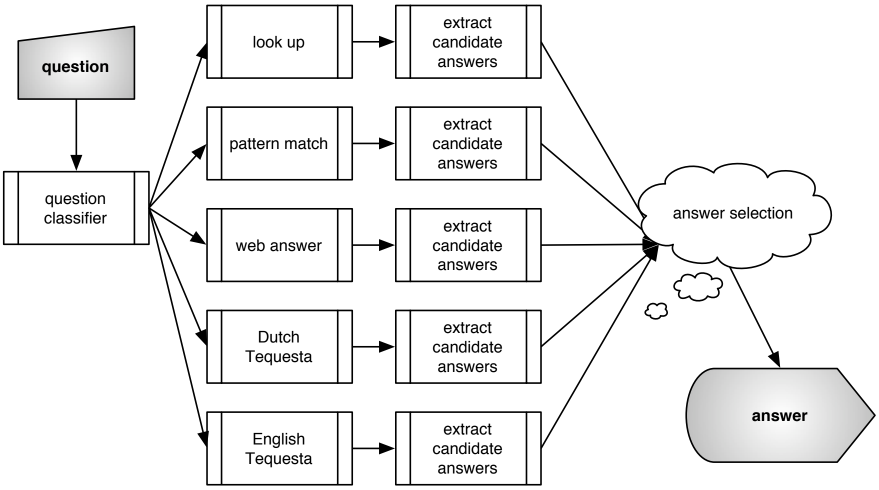

Figure 1: The University of Amsterdam's Dutch Question Answering System.

Question Answering Streams. We now provide a brief description of the five streams of our QA system: Table Lookup, Pattern Match, English Tequesta, Dutch Tequesta, and Web Answer.

The *Table Lookup* stream uses specialized knowledge bases constructed by preprocessing the collection, exploiting the fact that certain information types (such as country capitals, abbreviations, and names of political leaders) tend to occur in the document collection in a small number of fixed patterns. When a question type indicates that the question might potentially have an answer in these tables, a lookup is performed in the appropriate knowledge base and answers which are found there are assigned high confidence. For example, to collect abbreviation-expansion pairs we searched the document collection for strings of capitals in brackets; upon finding one, we extracted sequences of capitalized non-stopwords preceding it, and stored it in the "abbreviation knowledge base." This approach answered question such as:

Question	*84. Waar staat GATT voor?*
Knowledge Base	Abbreviations
Table Entry	GATT: Overeenkomst over Tarieven en Handel
Extracted Answer	**GATT**

For a detailed overview of this stream, see [3].

In the *Pattern Match* stream, zero or more Perl regular patterns are generated for each question according to its type and structure. These patterns indicate strings which contain the answer with high probability, and are then matched against the entire document collection. Here's a brief example:

Question	*2. In welke stad is het Europese Parlement?*
Generated pattern	`Europese Parlement\s+in\s+(\S+)`
Match	. . . voor het **Europese Parlement in Straatsburg**, dat . . .
Extracted Answer	**Straatsburg**

The *English Tequesta* stream translates the questions to English using Worldlingo's free translation service at `http://www.worldlingo.com/`. The auto-translated questions are then fed to *Tequesta*, an existing QA system for English developed at the University of Amsterdam [6]. The system uses the English CLEF corpus, and is extended with an Answer Justification module to anchor the answer in the Dutch collection.

The *Dutch Tequesta* is an adaptation of English Tequesta to Dutch and used as an independent stream, provided with the original Dutch newspaper corpus. The modifications to the original system included replacing (English) language specific components by Dutch counterparts; for instance, we trained TNT [1] to provide us with Part-of-Speech tags using the *Corpus Gesproken Nederlands* [7]; a named entity tagger for Dutch was also developed.

The *Web Answer* stream looks for an answer to a question on the World Wide Web, and then attempts to find justification for this answer in the collection. First, the question is converted to a web query, by leaving only meaningful keywords and (optionally) using lexical information from EuroWordNet. The query is sent to a web search engine (for the experiments reported here we used Google); if no relevant Web documents are found, the query is translated to English and sent again. Next, if the query yields some results, words and phrases appearing in the snippets of the top results are considered as possible answers, and ranked according to their relative frequency over all snippets. The Dutch named entity tagger and some heuristics were used to enhance the simple counts for the terms (e.g., terms that matched a TIME named entity were given a higher score if the expected answer type was a date). Finally, justifications for the answer candidates are found in the local Dutch corpus.

While each of the above streams is a "small" QA system in itself, many components are shared between the streams, including, for instance, an *Answer Justification* module that tries to ground externally found facts in the Dutch CLEF corpus, and a *Web Ranking* module that uses search engine hit counts to rank the candidate answers from our streams in a uniform way, similar to [4].

3 Runs

We submitted two runs for the Dutch question answering task: `uamsex031md` and `uamsex032md`. Both runs returned exact answers, and both combined answers from all streams, but differed slightly in the method of using the search engine hit counts for ranking the answers. The score of an answer was the product of the confidence measure produced by the stream generating the answer and the "Web Hit Count" measure, which equals the number of hit counts produced by Google for a query made up of the answer and keywords from the question. To prefer queries with words that do not occur frequently, we also calculated a "Query Value" measure: in `uamsex031md`, the query value was calculated using the word frequencies of the query words in the CLEF English and Dutch corpora, and in `uamsex032md` it was calculated using the Web hit count of the answer alone. Query values were used to normalize the Web Hit Count measure.

Here is a simplified example, in which the method used for `uamsex031md` produced better results (stream confidence level not displayed):

Question 115. *Waar bevindt zich de Klaagmuur?*		
Candidate Answer	Jeruzalem	Joyce
Generated Query	`Klaagmuur Jeruzalem`	`Klaagmuur Joyce`
Query Hit Count	793	26
Total Word Frequency	4.48e-05	1.85e-05
Candidate Hit Count	70700	3460000
Normalized Query Value (`uamsex031md`)	1.0	0.413
Normalized Query Value (`uamsex032md`)	0.02	1.0
Final Web Score (`uamsex031md`)	1.0	0.0135
Final Web Score (`uamsex032md`)	0.02	0.033

Shortly after the submission, we discovered a couple of implementation bugs that caused some of the Table Lookup stream answers to be incorrect. Below we also discuss two post-submission runs, `uamsex031md.fixed` and `uamsex032md.fixed`, which are identical to the submitted runs but with these implementation bugs fixed.

4 Results and Discussion

The following table shows the evaluation results of our CLEF 2002 submissions and the two post-submission runs described above. Beside the standard *Strict* and *Lenient* measures, we also evaluated our runs using more "generous" *Lenient, Non-exact* measure that accepts non-exact answers as correct.

	Strict		Lenient		Lenient, Non-exact	
Run	# correct answers	MRR	# correct answers	MRR	# correct answers	MRR
uamsex031md	78 (39%)	0.298	82 (41%)	0.317	96 (48%)	0.377
uamsex032md	82 (41%)	0.305	89 (44.5%)	0.335	102 (51%)	0.393
uamsex031md.fixed	84 (42%)	0.335	87 (43.5%)	0.352	100 (50%)	0.407
uamsex032md.fixed	88 (44%)	0.349	95 (47.5%)	0.375	107 (53.5%)	0.428

The run `uamsex032md` scored better than `uamsex031md`: as expected, normalizing web hit counts according to the distribution of words on the web yielded a more accurate ranking than normalization using corpus word frequencies. Also, the two runs with the fixed Table Lookup stream outperformed our official runs.

An error analysis of the questions which had a correct answer with incorrect document ID (i.e. those separating Strict and Lenient scores) revealed that answers with incorrect justifications did not necessarily come from external resources (the Web and English Tequesta streams); this suggests a local problem in our justification mechanism, rather than an inherent inability to justify externally found answers in the local corpus. Taking this into account, our 53.5% score in the table seems quite realistic.

It is interesting to see the increase in performance with the *Lenient*, *Non-exact* measure. Most of the non-exact answers that the system produced contained noise around the correct answer strings, e.g. “Jacques Delors. Met”, “Kim Il Sung. Japan” or “1989, heeft vooral in het oostelijke deel van Berl”, due to named entity extraction errors.

An initial analysis of the contribution of the different answering streams to the system’s overall performance suggests that every stream has its own strengths, that is, specific question types for which it provides correct answers with higher probability than other streams. The Web Answer stream, for example, seemed to perform better than other streams on questions for which the answer was a date; the Pattern and Table Lookup streams had very good performance on the specific (5-6) question types for which they were used. Every stream contributed some correct answers, so the total combined output of the system was better than any subsystem alone. E.g., out of the 200 questions, 54 (27%) were answered by the Table Lookup stream; of these, 26 answers (13% of the total answers) came solely from this stream. A further analysis of the performance of our streams on different question types will allow us to give each stream a confidence weight conditioned on question type, and thus to make the answer selection more informed, in ways similar to the approach adopted by BBN for TREC 2002 [9].

5 Conclusions and Future Work

We presented our multi-stream question answering system and the runs it produced for CLEF 2003. Running in parallel several subsystems that approach the QA task from different angles proved successful, as some approaches seem better fit to answer certain types of questions than others.

Our current ongoing work on the system is focused on extensions of the Table Lookup stream and the Web Answer stream. Future plans also include improvements of the voting mechanism between the answers provided by the different streams, and enhancing the system to support definition and list questions.

Acknowledgments

We want to thank Christine Foeldesi for her work on developing a named entity tagger for Dutch. Valentin Jijkoun and Gilad Mishne were supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. Maarten de Rijke was supported by grants from NWO, under project numbers 612-13-001, 365-20-005, 612.069.006, 612.000.106, 220-80-001, and 612.000.207.

References

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

[2] M. Banko et al. AskMSR: Question answering using the Worldwide Web. In *Proceedings EMNLP 2002*, 2002.

[3] V. Jijkoun, G. Mishne, and M. de Rijke. Preprocessing Documents to Answer Dutch Questions. In *Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence (BNAIC’03)*, To appear.

[4] Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Is it the right answer? exploiting web redundancy for answer validation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 425–432, 2002.

[5] D. Moldovan, S. Harabagiu, R. Girju, P. Morarescu, F. Lacatusu, A. Novischi, A. Badulescu, and O. Bolohan. LCC Tools for Question Answering. In Voorhees and Harman [8].

[6] C. Monz and M. de Rijke. Tequesta: The University of Amsterdam’s textual question answering system. In E.M. Voorhees and D.K. Harman, editors, *The Tenth Text REtrieval Conference (TREC 2001)*, pages 519–528. National Institute for Standards and Technology. NIST Special Publication 500-250, 2002.

[7] N. Oostdijk. The Spoken Dutch Corpus: Overview and first evaluation. In *Proceedings LREC 2000*, pages 887–894, 2000.

[8] E.M. Voorhees and D.K. Harman, editors. *The Tenth Text REtrieval Conference (TREC 2002)*. National Institute for Standards and Technology. NIST Special Publication 500-251, 2003.

[9] J. Xu, A. Licuanan, J. May, S. Miller, and R. Weischedel. TREC 2002 QA at BBN: Answer selection and confidence estimation. In Voorhees and Harman [8].