Multi-lingual Question Answering using OpenEphyra

Menno van Zaanen
Tilburg University
mvzaanen@uvt.nl

Abstract

In this article we describe our submission to the Dutch-English QA@CLEF task. We took the publicly available OpenEphyra question answering system, which is an open-source English question answering system. This was turned into a multi-lingual variant by translating questions from Dutch to English using Systran's online-translation system. The current approach has some known problems, for example, we do not distinguish between factoid, lists, and definition questions (all questions are treated as factoid questions), OpenEphyra does not provide support text for answers (text in the document surrounding the answer is used as support text), temporal restrictions and anaphora are not handled at all. The amount of modifications of OpenEphyra required to run the experiment were such that due to time constraints only one experiment could be submitted. The original idea behind this research was to investigate the impact of the quality of the question analysis. In particular, we are interested in the difference between the analysis on the question in the source language and the question in the target language.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; I.2.7 [**Natural Language Processing**]: Text Analysis

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Multi-lingual retrieval

1 Introduction

In 2007, we participated in the Dutch-English QA@CLEF task using the AnswerFinder system (van Zaanen and Mollá, 2007; Mollá and van Zaanen, 2005; van Zaanen et al., 2006). The aim of that research was to investigate the flexibility, configurability, and scalability of the AnswerFinder system. The experiment showed that AnswerFinder is flexible (in that it allows the necessary modifications), configurable (allowing for dynamic selection of parameters and algorithms in the different phases), and scalable (working on larger document collections) enough to be used for different projects. The experiment also showed that AnswerFinder is usable in a multi-lingual context, although several additional changes were needed to ensure useful results.

Currently, the main problem of using AnswerFinder is its reliance on the Connexor (Tapanainen and Järvinen, 1997) dependency parser. Licences for this parser are assigned per computer, which limits the portability of the system. In the meantime, the AnswerFinder project has finished and the licence has expired. This led to a largely reduced functionality of the system. While the framework is still useable, the shallow semantic representations (Mollá, 2006), which relies on output of Connexor, cannot be generated anymore.

With the limited usability of AnswerFinder in mind, there are two possibilities to resolve this problem. Firstly, we could re-implement the shallow representations based on the output of a different, non-commercial, dependency parser. This requires a thorough analysis of the output of the parsers and a mapping from the one representation to the other. Initial attempts at this have been made, but are currently not in a usable stage. Secondly, we could look for another question answering system that has a similar setup to AnswerFinder.

The original background of modifying AnswerFinder from an English-only question answering system into a multi-lingual one was to investigate the impact of the quality of the output of several phases. The idea is that when analysis is performed on (partially) incorrectly translated texts (in our case questions), the results will be worse than when that same analysis is performed on the original questions.

The research described in this article describes a first attempt at answering this question. We explain how we modify an existing English question answering system into a multi-lingual version by combining several existing components. Using this modified system, we participated in the Dutch-English QA@CLEF competition. Unfortunately, time constraints did not allow us to compare the results of question answering using analysis of source question against that of the target language. However, the system described here is ready to be used for this.

2 System components

The results of the Dutch-English QA@CLEF task are generated using a system that is created using several components. All of the components are freely available. Starting from the OpenEphyra open-source question answering system, we appended Systran's online machine translation system, which translated the Dutch questions into English, to allow OpenEphyra to answer English questions. In the future, we would like to perform question classification on the Dutch questions and incorporate these classes into the English question answering system.

2.1 Question answering system

We start building the multi-lingual question answering system using an existing question answering system. OpenEphyra[1] is an open-source question answering system. This system is based on Ephyra, which is developed by Nico Schlaefer and has participated in the TREC question answering competition (Schlaefer et al., 2006).

The system is written in Java and is highly configurable. In fact, the setup of the framework is very similar to AnswerFinder. It consists of a collection of factory classes (these generate objects with a certain interface) for all the phases in the system, which makes it very easy to select the wanted algorithm and also extending the system by adding a new algorithms is easy. As long as the algorithm provides the functionality required by the interface and the factory knows of it, the algorithm can be selected without rebuilding the system.

OpenEphyra needs to be modified in several ways before it can be used in this particular context. Firstly, by default, OpenEphyra searches for answers on the Internet. In QA@CLEF, the answers need to be found in a fixed set of documents. In 2008, for the Dutch-English task, the November 2007 dump of Wikipedia[2] (in HTML format), the 1994 collection of the Los Angeles Times (LA94) and the 1995 collection of The Glasgow Herald (GH95) were used.

[1] http://www.ephyra.info/

[2] http://www.wikipedia.com/

The indexing of all the documents is done using Indri[3], which is part of the open-source Lemur project[4]. Lemur is a toolkit for language modeling and information retrieval. Building the indices using Indri is straightforward. However, the documents need to be converted so Indri can find the correct XML tags to index. The LA94 documents are already in the correct format, but in the GH95 documents we need to inserted an XML "P" tag within the "TEXT" tag. The Wikipedia documents also need to be converted. We built documents in a format similar to those of the LA94 and GH95 documents. The filename of the Wikipedia page serves as the contents of the "DOCNO" and "DOCID" tags, the title of the page is used in the "HEADLINE" tag and the document contents are put within the "TEXT" tag (combined with a "P" tag). The indices generated by Indri are used by OpenEphyra directly.

Secondly, the input and output formats of the questions and answers in OpenEphyra are aimed towards the TREC competitions. However, the formats in used in CLEF are somewhat different. Starting from the questions in CLEF format, we removed all XML formatting. The plain text questions are then translated (as described in section 2.2). The translated questions are then inserted into an XML file in TREC format. The TREC format question file format contains similar information to the CLEF format with one main difference. Where in CLEF questions are organized by numbered topic, the TREC topics contain text. This text is used in OpenEphyra in the anaphora resolution module. Since in this experiment we do not have topic names, we have to turn anaphora resolution off (or implement a more complex algorithm that takes into account resolution based on words in the previous questions or answers). The file containing the questions in TREC format is fed to OpenEphyra, which results in an answer file in TREC format. This file is then converted into the final CLEF answer file. Essentially, this conversion is straightforward. The only problem is that in the TREC format, no support text (supporting the answer) is provided. We have solved this by retrieving the document in which the answer is found and identify the answer string in the text. The sentence around the answer is used as supporting text. Note that this does not necessarily provide correct support text, especially if the answer string occurs in the document multiple times.

Finally, we want to experiment with moving the analysis of the question from the target language (English) to the source language (Dutch). Normally, OpenEphyra performs question analysis. The idea is to perform the analysis on the original questions and insert this information into OpenEphyra directly, thus disabling the normal question analysis of the system. This is visualized in Figure 1. This figure shows the "standard" multi-lingual OpenEphyra layout. Following the dotted arrows as well (to the question analysis phase), the system is extended with the analysis on the source questions. Obviously, a new question analysis algorithm has to be incorporated in OpenEphyra that inserts the question information generated by the question analysis on the source questions. We have implemented an algorithm that reads question information from an external file, which allows us to perform the question analysis on the source questions first (before running OpenEphyra), write the results to file and use the contents of the file in OpenEphyra. This way, OpenEphyra does not explicitly have to know about the questions in the source language.[5]

2.2 Machine translation

To translate the questions from Dutch to English, we selected the Systran online machine translation system just like last year (van Zaanen and Mollá, 2007). We compared the quality of the output of several machine translations systems and based on these results, we selected Systran. Using the web interface provided[6], the Dutch questions are translated to English automatically.

Glancing over the translated questions, most questions are quite understandable even though native speakers of English might have phrased them otherwise. For example, "With which pianists he has cooperated?", "For which films he has written music?", or "Call two instruments which

[3] http://www.lemurproject.org/indri/

[4] http://www.lemurproject.org/

[5] In the end, the experiment that uses externally generated question information (based on the questions in the source language) were not submitted due to time restrictions.

[6] http://www.systranbox.com/

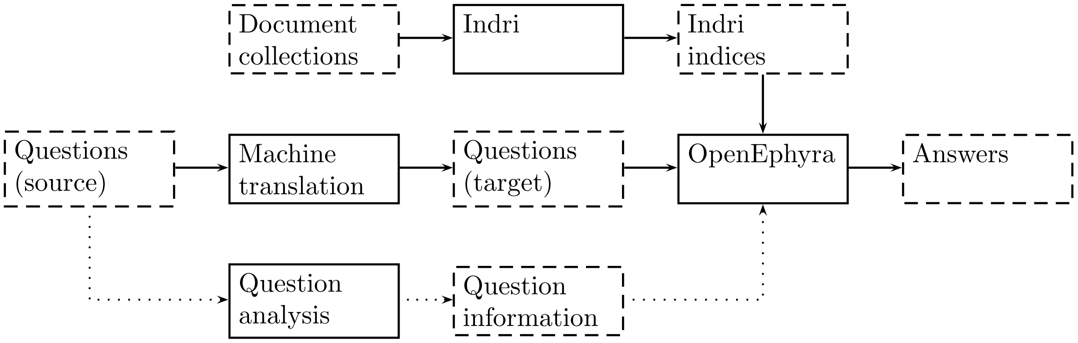

Figure 1: Layout of the extension of OpenEphyra for multi-lingual question answering.

are played on by Emerson." would probably be phrased differently. Other questions, however, are very unclear, such as "How did he come for living?" (from "Hoe kwam hij om het leven", meaning "How did he die?"), or "Where did he become herbegraven on 30 November 2002?" (from "Waar werd hij herbegraven op 30 november 2002?", meaning "Where was he re-buried on 30 November 2002?").

2.3 Alignment-Based Learning question classification

As question analysis tool, we wanted to use the question classifier that is based on Alignment-Based Learning (ABL) (van Zaanen et al., 2005). An ABL classifier takes a set of questions together with question classes for training. The questions are compared against each other, which identifies patterns in the questions. For each of the question classes, patterns are assigned. During testing, all of the patterns are matched against the new question. The question class that has most patterns matching is returned as the correct class.

Recently, a similar system has been applied to classifying musical pieces to their composer (Geertzen and van Zaanen, 1997). The system described here is similar to that of van Zaanen et al. (2005), but more specific patterns are generated. In the context of music classification the results are much better.

At the moment, we have not used this question classification system due to time constraints. The experiment that used the externally provided question classes took somewhat longer than the experiment using the internal question classification. Running and evaluating this approach and comparing the results against the experiments without external question classification is considered future work.

3 Results

Applying the system to the 200 questions provided, takes approximately ten hours from start to finish. This results in 200 answers, as all questions are considered of the factoid type. Of these answers 40 are NIL, indicating that the system cannot find an answer for those questions.

The overall accuracy of the system is 3.5%. Seven out of the 200 questions are answered correctly, two are inexact and nine unsupported. This means that 182 answers are incorrect.

The overall accuracy measure is very strict, taking into account that the system does not even try to identify and thus answer list or definition questions. Furthermore, the supporting text for an answer is selected in an ad hoc way.

Looking at the answers while considering the limitations of the system, more encouraging results are found. Firstly, if we do not require the system to provide correct support text (OpenEphyra does not generate support text) 16 answers out of the 200 are correct. This yields an accuracy of 8%. (Including the inexact answers as well, we get an accuracy of 9%.)

Secondly, if we take into account that the system only aims to generate answers for factoid questions, we see an accuracy of 4.375% (seven out of 160). If we then take include unsupported answers, we find an accuracy of 10% (16 out of 160). (There is one inexact factoid answer. Including that as well results in an accuracy of 10.625%). None of the ten list questions are correctly answered and only one of the 30 definition questions has an inexact answer, the rest is incorrect. Also, all 40 NIL questions are incorrect.

Thirdly, the system does not do anything about temporally restricted aspects of questions (or answers). As a results, as could be expected, all temporally restricted questions are answered incorrectly.

Finally, no anaphora resolution is performed, so it can be expected that all questions that contain anaphora are answered incorrectly. This is true except for one question, which is answered correctly, but does not have correct supporting text. The original question was "Hoe kwam hij om het leven?", which is translated as "How did he come for living?". The word "he" refers to "Jeremiah Clarke", which was mentioned in the previous question, but the system has no way of knowing this. The inexact answer to this question is "shot".

4 Conclusion

We built a multi-lingual question answering system using a collection of publicly available tools. The mono-lingual English OpenEphyra question answering system was extended using Systran's online machine translation system, translating Dutch questions to English. This simple system already generates encouraging results, given that several aspects were not treated at all. The system does not take into account temporally restricted questions, no anaphora resolution is performed, and only factoid questions are handled.

We wanted to investigate the impact of performing as much analysis on the original texts. In particular, we aimed at doing question analysis (classification) on the Dutch questions and compare the results against the system described in this paper, where question analysis was performed on the translated English question. Intuitively, one would expect the analysis on the texts in the original language to perform better.

Unfortunately, time restrictions did not allow us to evaluate our intuition regarding the quality of intermediate results with respect to the language of the source. We would have liked to perform question analysis on the Dutch questions and incorporate these results directly into the OpenEphyra system, disabling the internal question analysis algorithms (that work on the target language).

There are several possible directions for future work. Firstly, we would like to experiment with performing more analysis on the source language side. Question analysis is a first step in this direction, but query generation (used in the document selection phase) can perhaps also benefit from information extracted from the source questions. Secondly, we would like to improve the machine translation step. For example, finding named entities in the source language, including names and titles of books, songs, etc., and translating these separately may improve the translation quality. Systran consistently makes certain mistakes, for instance, using the word "call" instead of "name" (e.g. in "Call two instruments which are played on by Emerson."). We would like to investigate whether these mistakes can be recognized and corrected automatically. Finally, the current system does not handle anaphora at all. A new anaphora resolution algorithm needs to be implemented, which does not depend on topics having a name as in TREC.

References

Geertzen, J. and van Zaanen, M. (1997). Composer classification using grammatical inference. In Ramirez, R., Conklin, D., and Anagnostopoulou, C., editors, *Proceedings of the International Workshop on Machine Learning and Music (MML)*, pages 17–18.

Mollá, D. (2006). Learning of graph-based question answering rules. In *Proceedings of TextGraphs: the Second Workshop on Graph Based Methods for Natural Language Processing; New York:NY, USA*, pages 37–44.

Mollá, D. and van Zaanen, M. (2005). Answerfinder at TREC 2005. In *Proceedings of the Fourteenth Text Retrieval Conference (TREC 2005); Gaithersburg:MD, USA*, NIST Special Publication. Department of Commerce, National Institute of Standards and Technology. cd-rom.

Schlaefer, N., Gieselmann, P., and Sautter, G. (2006). The Ephyra QA system at TREC 2006. In TREC (2006).

Tapanainen, P. and Järvinen, T. (1997). A non-projective dependency parser. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97); Washington:DC, USA*, pages 64–71. Association for Computational Linguistics.

TREC (2006). *Proceedings of the Fifteenth Text Retrieval Conference (TREC 2006); Gaithersburg:MD, USA*, NIST Special Publication. Department of Commerce, National Institute of Standards and Technology.

van Zaanen, M., Augusto Pizzato, L., and Mollá, D. (2005). Question classification by structure induction. In *Proceedings of the International Joint Conferences on Artificial Intelligence; Edinburgh, UK*, pages 1638–1639.

van Zaanen, M. and Mollá, D. (2007). Answerfinder at QA@CLEF 2007. In Nardi, A. and Peters, C., editors, *Working Notes for the CLEF 2007 Workshop; Budapest, Hungary*.

van Zaanen, M., Mollá, D., and Pizzato, L. (2006). Answerfinder at TREC 2006. In TREC (2006).