Cross-Language Question Answering at the University of Helsinki

Lili Aunimo, Reeta Kuuskoski and Juha Makkonen
Department of Computer Science
University of Helsinki
P.O. Box 68, FIN-00014 UNIVERSITY OF HELSINKI
aunimo|rkuuskos|jamakkon@cs.helsinki.fi

Abstract

Tikka is a cross-language question answering system developed at the University of Helsinki for the purposes of the QA@CLEF 2004 evaluation campaign, *Tikka* was configured to answer Finnish questions using a text corpus in English, but it is designed so that it can be configured to work with any other languages as well. *Tikka* is the first general domain question answering system ever reported to have used Finnish. The question type classifier, the translator, the answer extractor and the answer scorer are the components of *Tikka* that are especially developed for question answering.

1 Introduction

Question answering (QA) is a task where the information need of the user is formulated as a natural language question and where the answer is given in natural language as well. In general, the length of the answer varies from one word to a couple of sentences, depending on the question. Also those situations where the system is not able to provide an answer should be detected. QA systems can be designed to work on a specific domain only (e.g. an aid for a company help desk [1, 2]), or they can be general purpose systems (e.g. TREC[1] and CLEF QA Tracks [2]). In general, question answering systems use unstructured text documents as their database, but, in addition, they can use lists of FAQs (Frequently Asked Questions) and structured databases. In general open domain QA systems, the Web can be used as a source of information. Much of the research on QA systems is concentrated in building systems for the CLEF and TREC evaluation campaigns. In these campaigns, the main database is newspaper text.

Cross-language question answering means that the question is expressed in another language than that in which the documents from which the answer is extracted are written. In this case, the user can use one language to search information from documents written in one or more other languages. This is useful, because it would be tiresome to write the question over and over again in many languages, and also because many users have a good passive knowledge of several languages, but their active knowledge is more restricted [3]. In our system, the questions can be expressed in Finnish and the document collection is in English. The system could be extended to handle questions and documents in other languages using the same methodology as presented in this paper. For the simplicity of presentation, we expect from here onwards that the questions and documents are in only one language, and that these languages are different. Cross-language QA is usually implemented either by first applying machine translation to the question and then passing it on to a monolingual QA system or by integrating cross-language processing into the QA system. Our approach is the latter one, because there is no reliable off-the-shelf machine translation software for Finnish. In addition, we expect to improve our results by using the

[1] http://trec.nist.gov/data/qa.html
[2] http://clef-qa.itc.it/2004/

original question as the basis of processing for as long as possible, because when translation is performed, the information content of the question is almost always altered.

In the following chapter we will descibe the overall architecture of our QA system. After that each of the main components of the system are described in detail. Section 3 descibes the processing of questions, that is, question classification and translation. In Section 4, the information retrieval component of our system is detailed. Answer processing, which consists of answer extraction pattern creation and instantiation and of answer selection and scoring, is described in Section 5. Section 6 is about evaluation and it presents our official results at QA@CLEF 2004. It also contains some discussion on the effects of translation on the overall performance of a QA system. Finally, Section 7 concludes.

2 System Architecture

The name of our QA system is *Tikka* (Woodpecker). It has three modules: *Question Processor*, *Information retrieval (IR) Engine* and *Answer Processor*. A system architecture is shown in Fig. 1. The figure also shows the configuration of *Tikka* for Finnish-English QA and for the document database used in the QA@CLEF evaluation initiative. The *Question* and *Answer Processors* are the modules which are especially developed for QA. The *IR Engine*, which is described in more detail in section 4, is a standard search engine. The *Question Processor*, which is described in section 3, first produces a syntactic parse of the question, then it classifies the question and finally it translates the relevant terms of the question. The *Answer Processor* first instantiates the answer extraction pattern prototypes with the translated words of the question. Then it applies the patterns to the documents retrieved by the *IR Engine* and finally it selects the best answer among the candidates extracted and gives it a confidence value. The *Answer Processing* module is described in detail in section 5.

When *Tikka* was used for the Finnish-English experiments of QA@CLEF, its document database consisted of 670 megabytes of newspaper text (The Glasgow Herald from 1995 and Los Angeles Times from 1994). Other external knowledge sources that the system used were the *MOT* dictionary software from *Kielikone Ltd.* [3], the functional dependency grammar parser from *Connexor Ltd.* [4] and a *Country and Capital Translation Database* extracted from the web site of *Statistics Finland* [5].

3 Question Processing

3.1 Question Classifier for Finnish

The question processing commences by determination of the question type. The possible types were defined already in Multisix corpus [5]: *date*, *location*, *measure*, *object*, *organization*, *other*, and *person*. In addition, CLEF 2004 introduced new two types: *manner* (answering how-questions), *abstraction* and *definition*. The last type was tagged in the evaluation corpus, and thus the number of types to be recognized was ten.

Obviously, the question type can often be determined just by looking at the question word. However, in Finnish this is not always a straight-forward task as the language is morphologically rich. Typically, instead of prepositions there are agglutinated morphemes denoting the inflected cases, and within a noun phrase, for example, the words comply to congruity, i.e., attributes follow the case of the head word. As a simple example, consider the following uses of 'who' (*kuka*):

[3] http://www.kielikone.fi/en/
[4] http://www.connexor.com/
[5] http://www.tilastokeskus.fi/index_en.html

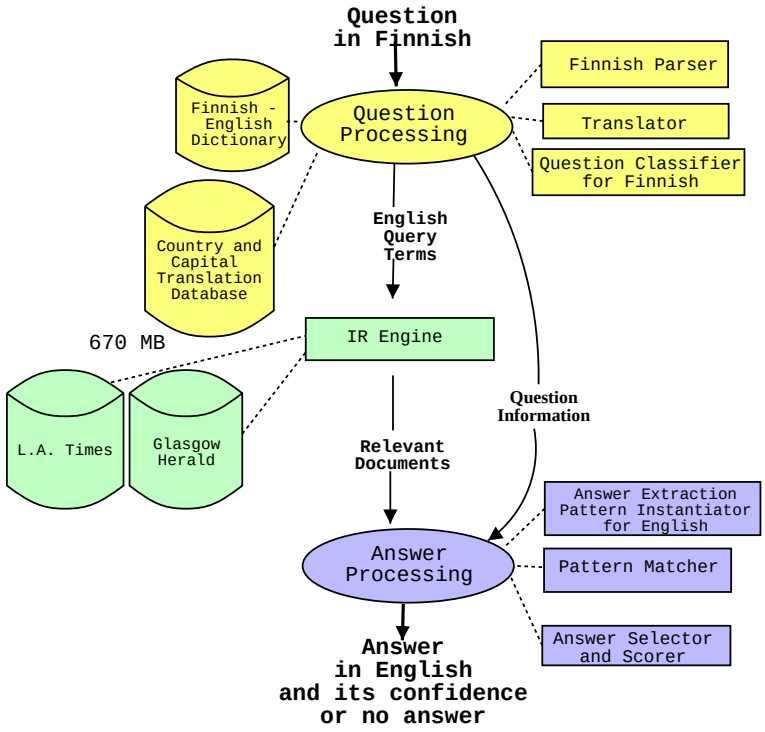

Figure 1: Architecture of Tikka and its configuration in the Finnish-English QA@CLEF.

English	Finnish	question word baseform + case
Who is that man?	*Kuka on tuo mies?*	kuka + NOMINATIVE
Who do you mean?	*Ketä tarkoitat?*	kuka + PARTITIVE
Who is he with?	*Kenen kanssa hän on?*	kuka + GENETIVE
Who do you think he is?	*Keneksi häntä luulet?*	kuka + TRANSLATIVE
Who has it?	*Kenellä se on?*	kuka + ADESSIVE
Who do you trust?	*Kehen luotat?*	kuka + ILLATIVE

There are 15 cases for each noun, adjective, pronoun and numeral in singular, and another 15 in plural. Furthermore, many morphemes produce changes also in the word body, and thus merely stripping morphemes at the end of the word is often of little avail. Without a morphological analysis it would be very difficult to take any further steps, because words are seldom used in their baseforms. We employ Connexor's functional dependency parser [4] for Finnish. Consider a sentence: *Minä vuonna se alkoi?* ('In what year did it start?'). It would be parsed as:

#	text	baseform	dependency	morphology
1	Minä	mikä	attr:>2	&A> PRON SG ESS
2	vuonna	vuosi	tmp:>4	&NH N SG ESS
3	se	se	subj:>4	&NH PRON SG NOM
4	alkoi	alkaa	main:>0	&+MV V ACT IND PAST SG3
5	?	?		

From this we see that, for instance, the pronoun *minä* is the essive of *mikä* (what) and that it is an attribute of the nominal head *vuosi* (year).

The time-related questions in the test corpus typically fell into one of three categories: a general 'when' (*milloin*, *koska*), a specific interval, e.g., 'what year|month|time' (*minä vuonna?*, *missä kuussa?*, *mihin aikaan?*) and a duration, 'how long' (*kuinka kauan*, *kauanko*, *miten kauan*, *kuinka pitkän aikaa*). The first two are date-questions and the last a measure-question.

Likewise, many measure-questions are somewhat straight-forward to recognize. The question is scanned for occurrence of quantity-related question words (e.g., *kuinka*|*miten moni*|*kauan*|*paljon*,

montako|moniko|paljonko|kauanko). Then there are 'what-is'-questions, such as *Mikä on Suomen väkiluku?* (What is the population of Finland?) The classification of the question relies in identifying the complement (population) as a measure-related word. Same technique is used with person, location, and organization related question: the type is based on classifying the object or the complement. Sometimes verbs are a helpful indicators of the question type.

The question classified in other-type, if the complement is a verb or if the complement or the object does not relate to person, location or organization. The manner-questions typically start with either *miten|kuinka* (how) or *millä tavoin|tavalla|keinoin* (in what manner|way). It has been difficult to identify object-questions as they vary considerably. Hence, we regard them as other-type.

3.2 Translator

Once the question has been classified, it is passed on to the *Translator*. It decides which of the words are translated, how to deal with proper names, homonyms and polysemous words and with words that have no translation in the dictionary. The *Translator* also decides which words are used in the query that is given to the *IR Engine*, and in answer extraction pattern prototype instantiation. For these decisions, it uses the syntactic parse tree of the question.

Once a question and its type is received by the *Translator*, it checks for country and capital names in the *Country and Capital Translation Database*. It contains 244 country and capital names in Finnish and their translations into English. The country and capital information is up to date as a new database is fetched from the web pages of *Statistics Finland* every once in a while. The version that we used in the CLEF evaluation exercise dates from 16.4.2004. This caused some problems, because the World has changed since 1994 and 1995 from where the CLEF newspaper text database dates. For example, two CLEF questions were about Yugoslavia, which our *Country and Capital Translation Database* naturally did not contain.

If the question contains a name that is in the database, it is given a translation and taken off from the list of words that will be passed on to the dictionary software. It is crucial that the proper names have been transformed into their baseforms before their existence in the database is checked because the database naturally does not contain any inflected proper names. For example, among the 34 country and capital names occurring in this year's questions, only 2 were uninflected.

After the *Country and Capital Translation Database* checking routine the translator determines which words are passed on to the dictionary software. All nouns are translated. If no translation is found, and the noun is a compound word, it is split into two parts both of which are used in the search from the dictionary. If there are more than two parts in the compound, then the last part forms the first search word and all the rest of the parts form the second search word. This is sensible, because quite often the preceding parts together are a modifier of the last part. For example (compound boundaries are marked with #): In *kori#pallo#joukkue* (basketball team) *kori#pallo* (basketball) modifies *joukkue* (team). This very coarse heuristic also has many counter-examples. One of them is *kulttuuri#pää#kaupunki* (Capital of Culture) where *kulttuuri* (culture) modifies *pää#kaupunki* (capital). In those cases where the noun is a compound word containing at least three parts and where the first part begins with a capital and ends with a hyphen, we split the word into dictionary search words from the hyphen, because the first part is most probably a proper noun and an uninflected modifier of the latter part and the latter part is the main part of the compound and it is inflected. For example in *Andrew-#pyörre#myrsky* (Hurricane Andrew) *Andrew* is a modifier for *pyörre#myrsky* (Hurricane). The proper noun could also contain several parts, for example *La# Scala -#ooppera#talo* (La Scala opera house), where *La# Scala* modifies *ooppera#talo* (opera house).

In addition to nouns, all adjectives that are attributes to nouns are translated. For example, in *How many Japanese students were there in the United States in 1990?*, *Japanese* is translated because it is a modifier of *students*.

If a word has no translation in the dictionary, and it looks like a proper name (begins with a capital and is not the first word of the question), its case is checked. If it is not nominative, but one of the other fourteen cases in which a noun can be, the baseform is passed on. Otherwise, the

original word in the question is passed on. This is because in the nominative case, no inflection is added to the proper name, while in the other cases, a suffix is added to the end of the word. In order to be able to use an inflected proper name as an English query term, we have to find its baseform.

The main reason for only translating nouns and their attributes is that the verbs used in the questions tend to be highly polysemious and they tend to have one or more homonymes. For example, in the case of this year's question number 40: *Who directed "Braveheart"?*, in Finnish *Kuka ohjasi elokuvan "Braveheart - Taipumaton"?* the verb *ohjata (to direct)* has 22 different senses in English, and only the seventh is the correct sense. However, the problem of polysemious words and homonyms also exists for nouns. For example, in this years question set, question 192 contained the word *laivasto* (navy), for which our dictionary software gave 3 different senses and 4 different translations (fleet, naval, forces and navy). If the different translations represent the same sense, they are often synonyms or regional variants. An example of synonyms: the translation candidates for *laulaja* are:

```
singer, songster, vocalist
```

which all represent the same sense according to our dictionary software. An example of regional variants: the translation candidates for *maanalainen* are:

```
metro, tube (br; the tube), underground (br; the underground), subway (yl am)
```

where *br* means British English and *am* means American English.

There are two main problems that could be studied further in the *Translator*. First, we should investigate whether query terms and answer extraction pattern prototype instantiation terms should be different. At the moment, the same terms are used for both.

The second area for further investigations is that of finding the correct translation or translations for a word in a given question. At the moment we take at most the two first translations and hope that the correct one is among these. Usually it is, because in general, the dictionary software lists the translation alternatives in the order of their frequency.

4 Information Retrieval

After the query terms have been selected, tey are given to the information retrieval engine. We used Managing Gigabytes (MG)[6] for IR task in *Tikka*. MG is an open source text indexing and retrieval engine developed as a joint venture of multiple Australian universities.

Prior to indexing, the documents were split so that each document was in its own file. The more fine-grained segmentation was not applied, since some of the answers to the training questions were not within one sentence, or even one paragraph. The files were then fed to MG for indexing. The contents of the documents were not otherwise preprocessed, although it might have enhanced the results, since the special characters caused some problems in retrieval. For instance, changing the dollar signs to corresponding strings might have been worthwhile.

The maximum number of retrieved documents was limited to one hundred, since we did not want the document sets to be processed in Answer Selection phase grow too large. In our experiments we noticed that if found at all, the document containing the correct answer was generally within the first 100. By default, MG was run in boolean query mode.

From our point of view, MG has some drawbacks. Firstly, it does not support phrase search or proximity constraints. This made it difficult to search for compound terms. It would also have been nice to be able to weight the terms according to their importance. For instance, one would have wanted to tell that the proper names occurring in the question are obligatory and they must be present in the retrieved documents, but other terms are less important. Now each of the terms were treated individually, and given the same relevance.

Especially with the questions that included proper names the boolean mode proved to work better than the ranked query. Since the query terms could not be weighted, the ranked query could sometimes give lots of irrelevant results. In the boolean mode at least the presence of the most fundamental terms can be required. Sometimes the query conditions were too strict, however, and

[6] http://www.mds.rmit.edu.au/mg

the result set became empty, in which case the mode was switched. This might cause the amount of the result document set to grow so large that the document with the correct answer could be left out of the set of 100 best and , hence, not be processed at all.

According to our experiments with the training data set, it seemed worthwhile to include also the corresponding adjective to the question as an alternative in case there was a name of a nation in the question. This is because the translations are in some situations more natural if the part of speech is altered. For example, question 29 in the test set was in English *What is the official German airline called?* The corresponding Finnish question is *Mikä on Saksan virallisen lentoyhtiön nimi?* Here *German* is an adjective, but *Saksan* is the genetive form of the noun *Saksa* (Germany).

Another motivation for adding the corresponding adjectives/nouns is the fact that even within one natural language, both of these expressions occur in sentences that have the same meaning. For instance, question 90 in the training set was *How many people in U.S. do not have health insurance?*, where *U.S.* is a noun. The correct answer to it was 37 million, which existed in the following snippet: *... the existing system, which leaves 37 million Americans without health insurance and ...* There the triggering term is *American*, which is an adjective.

The expansion of the query terms with synonyms would probably have improven the results. The disambiguity of the query terms, especially in bilingual question answering task, enlargens the expansion term candidate set notably, however. Some proper names could have quite easily be expanded, though, such as United States, which might have been worth expanding with terms *US, America* and *American*, as discussed above.

The most important terms in the query seemed to be the proper nouns, as one might expect. After that came the common nouns, possibly expanded with their synonyms. Next to the common nouns were verbs, outside of some verbs that were so common that they didn't actually mean anything (such as *do, be*). The least important group of words were generally the adjectives, though there were some questions in which the adjectives were very significant, for instance in question 79: *What is the highest active volcano in Europe?*. This has been taken into account in query term selection, as was described in section 3.2.

The seach results are passed onward to Answer Selection module for the execution of the next phrase, the answer extraction.

5 Answer Processing

5.1 Answer Extraction Patterns

Answer extraction pattern instantiation is the first step in *Answer processing*. This is done by creating instances of pattern prototypes. Each question type has a set of pattern prototypes that have been induced from the 1994 L.A. Times and the 1995 Glasgow Herald using the Multisix Corpus [5]. The pattern prototypes have slots where translated words from the question are inserted in order to form pattern instances.

Tikka contains pattern prototypes for six question types. They are: *date, definition, location, measure, person* and *other*. Based on the question types in the Multisix Corpus [5], we could have developed pattern prototypes also for the classes *object* and *organization*. However, we picked the most common categories for pattern prototype development and left the rest for future development. *Other* is a class where we classify all those questions that do not belong to the other five classes. In addition, the *CLEF-2004 Question Answering Track Guidelines* [7] contained classes *abstraction* and *manner*, but we did not develop pattern prototypes for these since we had no training material.

Below are 3 examples of the 11 instantiated location patterns for question 116 *Where is the Reichstag?*

```
[Ii]n (([A-Z][a-z]+ ){1,5}[A-Z][a-z]+), a [a-z]* [a-z]+,[^a-zA-Z0-9]+Reichstag[,\.]
 at ([A-Z][a-z]+,? ([A-Z][a-z]+)?), [^\.\?\!0-9\",]* Reichstag
```

[7] http://clef-qa.itc.it/2004/guidelines.html

Class	Patterns
date	3
definition	3
location	18
measure	22
other	5
person	16
total	70

Table 1: Question type classes and number of prototype patterns in each class.

```
Reichstag[^<>\?\!\;\.]*\.?[^<>\?\!\;\.]* in ((the )?([A-Z][a-z\']+,? ){0,3}[A-Z][a-z\']+)[\.,][^0-9]{2}
```

The prototypes of these patterns are identical to the instantiated patterns, except that the word *Reichstag* is replaced with a wildcard denoting any noun from the question. The third pattern is the one that matched both of the answers that were found. Both of the answers are *Berlin*, and here is their context:

Two matches for question 116:
WORKERS lower a giant panel of cloth over the entrance to the Reichstag in Berlin, helping Hungarian artist Christo to fulful a dream of 24 years.
Reichstag in Berlin, he
He will use 160 assistants to wrap the Reichstag with 90,000 square yards of a silver propylene fabric, chosen "because it fits with the building, the heaven and light in Berlin." Christo
Reichstag with 90,000 square yards of a silver propylene fabric, chosen "because it fits with the building, the heaven and light in Berlin."

Table 2: The two text snippets that were matched for question 116. Above is a bigger window of text and below is the exact text snippet that was matched by the pattern.

The answer pattern prototypes consist of regular expressions and of slots for proper names and other words that have been picked from the question. The answer pattern prototypes do not contain any syntactic or morphological information at the moment. Table 1 lists all the pattern classes and the number of prototype patterns that each class contains. In future research, it would be interesting to incorporate at least part of speech information into the patterns. Examples of pattern instances that are derived from the same location pattern prototype:

```
the city of ([^ ,\.\?\!0-9]+), Mike Kelley[^\.\?\!0-9]*
the town of ([^ ,\.\?\!0-9]+), Mike Kelley[^\.\?\!0-9]*
```

In the above example, the word *kaupunki* has two translations, *city* and *town*, and the pattern prototype is expanded with both.

Another example:

```
PROPER NAME[^,\.\?\!0-9]* TITLE,? [^A-Z]*(([A-Z][a-z]+[ -])*[A-Z][a-z]+)
```

In the above person pattern prototype the slots for *PROPER NAME* and *TITLE* are filled with words from the question. For example, in the question 2 from 2003, *Kuka on YK:n pääsihteeri?, Who is the head of the United Nations?*, the slot for *PROPER NAME* is filled by *UN*, *United Nations* and *UN (United Nations)*. The slot for *TITLE* is filled by *Sectretary General* and *secretary-general*. When all these instantiations are combined, we end up with 6 different pattern instances. The different variations for the slots except for the combination *UN (United Nations)* are retrieved from the dictionary. For all acronyms that have the longer form listed in the dictionary, the system performs the same type of expansion as for *UN*.

5.2 Answer Selection and Scoring

Answer selection is based on frequency, which means simply that among the answer candidates, the answer that appears most often is selected. If there are several answer candidates with the

same frequency, the one appearing first in the results retrieved by the *IR Engine*, is selected. This is a reasonable approach, because the *IR Engine* search results are ranked in the order of relevance.

Confidence measure generation is a function of both the total number of candidates retrieved and of the frequency of the selected candidate. This function is illustrated as an area plot in Figure 2. In *Tikka*, the frequencies and numbers of different candidates are discrete and not continuous as shown in the figure. The confidence score is 1, if the number of different candidates is a number between 1 and 5, or if the number of different candidates is a number between 6 and 14 and the frequency of the candidate is greater than 1 (the area marked with tiles in figure 2). The confidence score is 0.5 if the number of different candidates is between 6 and 10 and the frequency is 1 (the area marked with diagonal lines in figure 2). The confidence score is 0.25 if the number of different candidates is between 11 and 14 and the frequency is 1, or if the number of different candidates is over 14 (the area marked blank in figure 2). All those answers that we detected as not having an answer in the text database (answers of type *NIL*) had a confidence score of 0. Detecting the degree of confidence for answers of type *NIL* is a goal for future research.

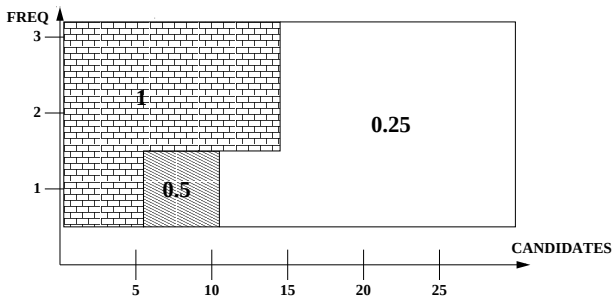

Figure 2: Area plot of the confidence score of *Tikka* as a function of candidate frequency and total number of different candidates.

The Table 3 lists the number of occurrences of each confidence measure and the number of correct answers in these classes. Only answers that are not *NIL* are considered.

Confidence	CLEF 2004 (official)		CLEF 2003 (unofficial)	
	Occurrences	Correct	Occurrences	Correct
1	27	5 (one inexact)	51	29
0.5	2	0	3	0
0.25	5	0	14	4
0	0	0	0	0

Table 3: Number of occurrences of different confidence measures and number of correct answers. No figures are given for the NIL-answers.

As can be seen from the table 3, the confidence function should have been more strict, i.e. the score 1 should have been given to fewer answers. However, the confidence function depends heavily on the data and questions at hand and on how well the answer extraction patterns match to that data. We trained *Tikka* with questions and answers from QA@CLEF from 2003, and it seems that the answer extraction pattern prototypes were too specific to those answers. With the 2003 questions we got 132 NIL answers, but with this years material, the number of NIL answers was 159. The distribution of confidence scores from 2003 is shown in table 3.

6 Evaluation

6.1 Our Results at QA@CLEF

Our results in the 2004 Finnish-English QA@CLEF are shown in figure 4.

	Absolute numbers	Percentage
Accuracy	21/193	10.88
Accuracy of factoid questions	20/173	11.56
Accuracy of definition questions	1/20	5
Number of NIL answers	159/193	82.38
Accuracy of NIL answers	17/159	10.69
Confidence-weighted score		4.65

Table 4: Our results at 2004 QA@CLEF.

We had 21 right answers, among which 20 were factoid questions and 1 was a definition question. One of our answers was inexact and there were no unsupported answers in our answer set.

6.2 Inter-Translator Agreement

The questions for Finnish-English QA were translated from English. The assessor of the evaluation campaign compared the English questions against the results given by *Tikka*. However, the translation process is not straightforward, because for most questions, there seem to be as many translations as there are translators. In addition, not all questions are sensible when translated. For example, the question 86 (*What does a luthier make?*) became pointless in Finnish, because our word for luthier (*soitinrakentaja*) tells what a luthier does. Another example of the influence of translation on the questions is question 85 *What did the artist Christo wrap up?*. This can be translated in two ways which have a completely different meaning due to the ambiguity of the verb *to wrap up*. *To wrap up* can be translated as denoting concrete wrapping up, which was the correct meaning according to the correct answer, which is that *The artist Chisto wrapped up the Reichstag in silver fabric tied with blue rope*. The other meaning of *to wrap up* is an abstract one, and it means finishing something. The translations 1 and 2 translated *to wrap up* with its concrete sense *paketoida*, but translation 2 has the abstract sense *saattaa päätökseen*. We did two more translations of the English questions by translators who had not seen the official translation in order to measure the difficulty of the translation task and the inter-translator agreement rate. The amount of inter-translator agreement is illustrated in Table 5. *Translation 1* is the official translation where the errors have been corrected [8].

	Absolute numbers	Percentage
All three translations:	51/200	25,5
Translation 2 and 3:	86/200	43
Translation 1 and 2:	74/200	37
Tarnslation 1 and 3:	69/200	34,5

Table 5: Number of indentically translated questions.

The most common translator disagreement types are lexicografic disagreement, word order disagreement and disagreement in the use of conventions. Lexicographic disagreement means a different choice of words where the words are synonyms or semantically very closely related. For example: *manufacture* translated as *valmistaa* or *tuottaa*. Word order disagreement means that the words in the question are in a different order. For example: *Missä on Hyde Park? (Where is Hyde Park?)* and *Missä Hyde Park on? (Where Hyde Park is?)*. Disagreement on the use of conventions means that there are many, equally correct, different conventions on how to express a concept. For example, there are several conventions for expressing names of movies that have originally appeared in another language than Finnish. For example, the question 175 is about the movie *Nikita*. *Nikita* was translated into Finnish in three different ways: *Nikita*, *elokuva "Tyttö nimeltä Nikita"* and *Nikita (La Femme Nikita)*. The translation of names of movies is problematic because some movies have an official translation into Finnish and some don't. In the case of *Nikita*, there were two official translations, *Nikita* and *Tyttö nimeltä Nikita*. Proper names are

[8] http://clef-qa.itc.it/2004/down/clef04-test-FI-EN-correct.txt

often typed, *elokuva "Tyttö nimeltä Nikita" (movie "Nikita")*, because then the type (*movie*) gets inflected and there is no need to inflect the proper name. One convention in expressing movie names is that of first writing the official translation in Finnish and then adding the name of the original movie in parenthesis after it, as in *Nikita (La Femme Nikita)*. It will be interesting to compare the QA results with translations 1, 2 and 3 once we have the correct answers for this year's questions.

7 Conclusions and Future Work

To the best of our knowledge, the work presented in this paper is the first time cross-language QA has been done using Finnish as a source language. Altogether, there has been very little work on any type of QA for Finnish. Keeping this in mind, it was interesting to get the system up and running and to observe that it could answer 10,88 % of the questions presented to it correctly.

Due to the very different nature of Finnish in comparison to any of the other languages participating in the QA@CLEF, special attention has been paid to question translation and to the effects of the translation phase to the overall performance of the system. This is also a subfield on which we plan to focus our attention in the future.

Another interesting subfield is that of answer extraction patterns. We plan to study carefully which patterns matched well and which didn't and to find out the reasons for this. We are also planning to investigate the use of POS tags and possibly surface syntactic tas in the answer extraction patterns. The results obtained in this evaluation showed that by developing further the question and answer processing modules, as well as by tuning the IR engine more carefully, the performance of *Tikka* is very likely to improve.

References

[1] L. Aunimo, O. Heinonen, R. Kuuskoski, J. Makkonen, R. Petit, and O. Virtanen. Question answering system for incomplete and noisy data: Methods and measures for its evaluation. In *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR 2003)*, pages 193 – 206, Pisa, Italy, 2003.

[2] S. Busemann, S. Schmeier, and R. G. Arens. Message classification in the call center. In *Proceedings of 6th Applied Natural Language Processing Conference*, Seattle, Washington, USA, 2000.

[3] J. Gonzalo. Scenarios for interactive cross-language retrieval systems. In *Proceedings of the Workshop 1: Cross-Language Informaion Retrieval: A Research Roadmap Workshop held at the 25th Annual International ACM SIGIR Conference*, Tampere, Finland, aug 2002.

[4] T. Järvinen and P. Tapanainen. A dependency parser for english. Technical Report TR–1, Department of General Linguistics, University of Helsinki, 1997.

[5] B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Penas, V. Peinado, F. Verdejo, and M. de Rijke. The Multiple Language Question Answering Track at CLEF 2003. In C. Peters, editor, *Working Notes for the CLEF 2003 Workshop*, Trondheim, Norway, aug 2003.