Disambiguation Strategies for Cross-language Information Retrieval

Djoerd Hiemstra and Franciska de Jong

Centre for Telematics and Information Technology
University of Twente, Enschede, The Netherlands
{hiemstra,fdejong}@cs.utwente.nl

Abstract. This paper gives an overview of tools and methods for Cross-Language Information Retrieval (CLIR) that are developed within the Twenty-One project. The tools and methods are evaluated with the TREC CLIR task document collection using Dutch queries on the English document base. The main issue addressed here is an evaluation of two approaches to disambiguation. The underlying question is whether a lot of effort should be put in finding *the* correct translation for each query term before searching, or whether searching with more than one possible translation leads to better results? The experimental study suggests that the quality of search methods is more important than the quality of disambiguation methods. Good retrieval methods are able to disambiguate translated queries implicitly during searching.

Keywords: Cross-Language Information Retrieval, Statistical Machine Translation.

1 Introduction

Within the project Twenty-One a system is built that supports Cross-language Information Retrieval (CLIR). Cross-language retrieval supports the users of multilingual document collections by allowing them to submit queries in one language, and retrieve documents in any of the languages covered by the retrieval system. On the example of Dutch queries on an English document collection, this can be achieved by: *off-line document translation*: translating English documents into Dutch, then indexing in Dutch; *off-line index translation*: indexing English documents in English, then translating the resulting index into Dutch; *on-line query translation*: indexing English documents in English and translating Dutch queries on the fly into English. The latter method is preferred when the former two are impractical. Query translation is envorced in environments where it would be impossible to produce translations for all documents in the document base and/or to produce translated indices for each language. Document translation has the major advantage that it is possible to present the user a high quality preview of the retrieved documents. Translating documents after they are retrieved, as offered by some web search engines, does not suffice because it does not help the users to identify material that they might want to have translated. Since it presupposes that the user has already found the relevant document in its original foreign language, it fails to support exactly that part of a search in a multilingual environment which is the most difficult one: to formulate a query which will then take the user to the foreign language document of interest.

The Twenty-One project has a clear target domain. It focuses on disclosing literature on sustainable development in four languages: Dutch, English, French and German. The project also has a strong focus on the disclosure of paper documents which have to be scanned and converted to an electronical format by optical character recognition software. A third focus is on natural language processing in the four supported languages. At indexing time noun phrases are recognised and used as complex index terms. As the Twenty-One domain is limited and as heavy preprocessing and storage of scanned documents has to be reckoned with anyhow, this is a classic case for the document translation approach. The Twenty-One system[1] uses sophisticated translation software for the disambiguation of index terms in context. If a word has more than one possible translation it is called ambiguous, e.g. the English word *plant* has two possible French translations *plante* for the sense of 'vegetation' and *usine* for the sense of 'factory'. The term disambiguation is used in two ways in this paper. Disambiguation refers to the process of choosing the best translation. However, disambiguation might also refer to the estimation of probabilities for each possible translation. The disambiguation process might for instance assign a probability of 0.8 to *plante* and 0.2 to *usine*. The probabilities can be used to identify the most probable translation, but, if the query translation approach is taken, they might also be used during retrieval to weight each possible translation. Currently disambiguation in Twenty-One can be pursued in four ways:

- using existing machine translation software (LOGOS)
- selection of the preferred translation from a machine readable dictionary (Van Dale)
- the use of domain specific dictionaries that are automatically generated on the basis of statistically processed parallel corpora (suited for specific applications only)

[1] Twenty-One was the first on-line text retrieval system supporting CLIR in Europe: http://twentyone.tpd.tno.nl/21demomooi/

– disambiguation on the basis of the frequency of noun phrases in the document collection

In this paper the different disambiguation strategies of the Twenty-One system will be evaluated. The paper addresses the question which strategy results in the best retrieval performance, but it also addresses the question if disambiguation is necessary at all. New probabilistic retrieval techniques that are developed at the University of Twente are able to disambiguate translated queries implicitly during searching.

This paper is organised as follows. Section 2 explores possibilities for the comparison of the "document translation" approach with the "query translation" approach. Section 3 introduces three basic methods for query translation. Section 4 addresses heuristics and statistics for disambiguation if the query translation approach to cross-language retrieval is followed. Section 5 discusses the setup of our experiments and experimental results. Finally, section 6 contains concluding remarks. Technical details of the probabilistic retrieval model can be found in the appendix of this paper.

2 Empirically comparing document translation with query translation

In the introduction three important advantages of document translation were mentioned. Firstly, it can be done off-line. Secondly, if a classical machine translation is used, it is possible to present the user a high quality preview of a document. Thirdly, there is more context available for lexical disambiguation which might lead to better retrieval performance in terms of precision and recall.[2] For several types of applications, the first and second advantage may be a good reason to choose for document translation. The third advantage seems quite plausible and was hypothesised in a number of early publications on cross-language retrieval, e.g. Oard and Dorr [16], Hull and Grefenstette [10] and Kraaij [11].

Does the document translation approach to cross-language retrieval using classical machine translation *really* lead to better retrieval performance than the query translation approach using a machine readable dictionary? A recent experimental study by Oard [15] suggests it does. However, for a number of reasons it is very difficult to answer this question on the basis of empirical evidence. A first problem is that in the query

[2] Precision is the fraction of the retrieved documents that is actually relevant and recall is the fraction of the relevant documents that is actually retrieved.

translation approach, searching is done in the language of the documents while in the document translation approach searching is done in the language of the query. But it is a well known fact that information retrieval is not equally difficult for each language. A second problem is that, for a sound answer to the question, we need a machine translation system and a machine readable dictionary that have exactly the same lexical coverage. If the machine translation system misses vital translations that the machine readable dictionary does list, we end up comparing the coverage of the respective translation lexicons instead of the two approaches to cross-language retrieval. Within the Twenty-One project we have a third, more practical, problem that prevents us from evaluating the usefulness of the used translation system (LOGOS) against the usefulness of the machine readable dictionaries available within the project (Van Dale). The Van Dale dictionaries are entirely based on Dutch head words, but translation from and to Dutch is not supported by LOGOS. These considerations urge us to rephrase the the issue into a more manageable question.

A first, manageable, step in comparing document translation with query translation might be the following. What is, given a translation lexicon, the best approach for query translation: using one translation for each query term (i.e. explicit disambiguation) or using all possible translations? Picking one translation is a necessary condition of the document translation approach. For query translation we can either use one translation for searching, or more than one. The question one or more translations also reflects the classical precision / recall dilemma in information retrieval: picking one specific translation of each query term is a good strategy to achieve high precision; using all possible translations of each query term is a good strategy to achieve high recall.

3 Methods for query translation

As said in the previous section one of the issues dealt with in this paper is comparing cross-language information retrieval using one translation per query term with retrieval using more than one translation per query term. We will report the results of retrieval experiments using the Dutch queries on the English TREC cross-language task collection. A Dutch query will be referred to as the source language query; the English query will be referred to as the translated query. The experiments can be divided into three categories:

1. query translation using one translation per source language query term

2. query translation using unstructured queries of all possible translations per source language query term
3. query translation using structured queries of all possible translations per source language query term

3.1 Using one translation per query term

If only one translation per query term is used for searching, the translation process must have some kind of explicit disambiguation procedure. This procedure might be based on an existing machine translation system, or alternatively, on statistical techniques or heuristics. After disambiguation, the translated query can be treated the way a query is normally treated in a monolingual setting. A 'normal' monolingual setting in this context is retrieval on the basis of a statistical 'bag-of-words' model like e.g. the vector space model [20] or the classical probabilistic model [18]. The unstructured queries mentioned in the next section will also refer to the use of a bag-of-words model. Instead of the vector space model or the classical probabilistic model we will use a new model, called the linguistically motivated probabilistic model of information retrieval, which is described in the appendix of this paper.

Figure 1 gives an example of an English query {*third*, *world*} that is used to search a French collection. Although both *third* and *world* might have more than one possible translation, the system has to pick one of them.

$$\begin{array}{l} \{\textit{third},\ \textit{world}\} \\ \big\downarrow \ \text{dictionary lookup and disambiguation} \\ \{\textit{tiers},\ \textit{monde}\} \end{array}$$

Fig. 1. using one translation per query term

In section 4 a number of heuristics and statistics for disambiguation will be explored. As explained in section 2 we will not be able to actually use machine translation for disambiguation. It is however possible to define an upper bound on what is possible with the one-translation approach by asking a human expert to manually disambiguate the output of the machine readable dictionary. We hypothesise that query translation using a machine translation system with the same lexical coverage as the machine readable dictionary will not result in better retrieval performance than query translation using the manually disambiguated output of the same dictionary.

3.2 Using unstructured queries

If more than one translation per source language query term is used for searching we might still treat the translated query as a bag-of-words. As we will see in section 5 the way of weighting the possible translations is crucial for unstructured queries. In particular it is important to normalise the possible translations in such a way that for each source language query term the weights of possible translations sum up to one. Not using normalisation will make source language query terms with a lot of possible translations unintentionally more important than source language query terms that have less possible translations.

Figure 2 again gives the example of an English query {*third*, *world*} that is used to search a French collection. It is assumed that the English term *third* has two possible French translations: *tiers* and *troisième* and that the English term *world* has three possible translations: *monde*, *mondial* and *terre*. Instead of selecting one translation we might use all possible translation to search the document collection.

{*third*, *world*}
↓ dictionary lookup
(*tiers*, *troisième*, *monde*, *mondial*, *terre*)

Fig. 2. translation using an unstructured query

The result of figure 2 could be used directly for searching the French collection (see **run2a** in section 5), but this would make the term *world* in the source language query more important (because it has more possible translations) than the word *third*. Normalisation of the possible translations might therefore be used to make the contribution of *third* as high as the contribution of *world*. In this case the possible translations of *third* are reweighted to 0.5 and the possible translations of *world* to 0.33 (see **run2c** in section 5). If one of the possible translations of one source language query term is more probable than the other(s), this possible translation might be weighted higher than the other(s) while keeping the normalisation in tact.

3.3 Using structured queries

If all possible translations are treated as one bag-of-words we ignore the fact that a document containing one possible translation of each source language query term is more likely to be relevant than a document containing all possible translations of only one source language query term. The boolean model or weighted boolean models (see e.g. [20]) can be used to retrieve only those documents that contain a translation of all or most of the source language query terms [9]. Disjunction can be used to combine possible translations of one source language query term. Conjunction can be used in a way that the translated query reflects the formulation of the source language query.

$$\begin{array}{c} \{\textit{third},\ \textit{world}\} \\ \big\downarrow \text{ dictionary lookup} \\ \{(\textit{tiers} \cup \textit{troisi\`eme}),\ (\textit{monde} \cup \textit{mondial} \cup \textit{terre})\} \end{array}$$

Fig. 3. translation using a structured query

Figure 3 again gives the example of an English query {*third*, *world*} on a French document collection. The structured query reflects the possible translations of the source language query terms in an intuitive way. The structured query weighting algorithm implicitly normalises the possible translations in a disjunction. Explicit normalisation as done for unstructured queries is no longer necessary. Structured queries are generated automatically by the translation module and may take the probabilities of possible translations into account. Technical details of the algorithm can be found in the appendix.

4 Heuristics and statistics for disambiguation

This section lists a number of information resources that can be used to identify the proper translation or proper translations of a query term. The section briefly describes information that is explicitly or implicitly in the dictionary and information from other sources like parallel corpora and the document collection itself.

4.1 Dictionary preferred translation

The VLIS lexical database of Van Dale Lexicography list for each entry explicitly one *preferred* translation which is considered the most commonly used one. Replacing each query term with the preferred translation is a simple, but possibly effective, approach to cross-language retrieval.

4.2 Pseudo frequencies

The Van Dale database contains also explicit information on the sense of possible translations. Some Dutch head words carry over to the same English translation for different senses. For example the Dutch head word *jeugd* may be translated to *youth* in three senses: the sense of 'characteristic', 'time-frame' and 'person'. The 'person' sense has a synonym translation: *youngster*. As *youth* occurs in the dictionary under three senses and *youngster* under one sense, we assign *youth* a weight that is three times as high as the weight for *youngster*. The assumption made by weighting translations is that the number of occurrences in the dictionary may serve as rough estimates of actual frequencies in parallel corpora. In other words: the number of occurrences in the dictionary serve as *pseudo frequencies*. Ideally, if the domain is limited and parallel corpora on the domain are available, weights should be estimated from actual data as described in section 4.3.

4.3 Frequencies from parallel corpora

The Twenty-One system contains documents on the domain of sustainable development. Translation in Twenty-One is done using a general purpose dictionary (Van Dale) and a general purpose MT-system (LOGOS), but these resources are not very well suited for domain-specific jargon. Domain-specific jargon and its translations are implicitly available in parallel corpora on sustainable development. Translation pairs can be derived from parallel corpora using statistical co-occurrence by so-called word alignment algorithms. Within the Twenty-One project word alignment algorithms were developed that do the job in a fast and reliable way [6]. Domain specific translation lexicons were derived from Agenda 21, a UN-document on sustainable development that is available in most of the European languages including Dutch and English.

For the experiment we merged the automatically derived dictionary with the Van Dale dictionary in the following way. For each entry, we added the pseudo frequencies and the real frequencies of the possible translations. Pseudo frequencies are usually not higher than four or five, but the real frequencies in the parallel corpus may be more than a thousand for frequent translation pairs. Adding pseudo frequencies and real frequencies has the effect that for possible translations that are frequent in the corpus the real frequencies will be important, but for translations that are infrequent or missing the pseudo frequencies will be important.

Translation pairs that have a frequency of one or two in the parallel corpus may-be erroneously derived by the word alignment algorithm. If, however, such an infrequent translation pair is also listed in the machine readable dictionary, then the pair was probably correct. Therefore we added a bonus frequency of three to each possible translation that is both in the corpus and in Van Dale.

4.4 Context for disambiguation

The techniques introduced so far do not resemble techniques that are actually used in machine translation systems. Traditionally, disambiguation in machine translation systems is based on (syntactic) context of words. In this section a statistical algorithm is introduced that uses context of the original query words to find the best translation. The algorithm uses candidate noun phrases extracted from the document base to disambiguate the from the query. Noun phrases were extracted using the standard tools as used in the Twenty-One system: the Xerox morphological tools and the TNO parser. The noun phrases were sorted and then counted, resulting in a list of unique phrases with frequency of occurrence.

The introduction of noun phrases (or any multi-word expression) in the translation process leads to two types of ambiguity: sense ambiguity and structural ambiguity. Figure 4 gives an example of the French translation chart of the English noun phrase *third world war*. Each word in this noun phrase can have several translations that are displayed in the bottom cells of the chart, the so-called sense ambiguity. According to a list of French noun phrases there may be two candidate multi-word translations: *tiers monde* for the English noun phrase *third world* and *guerre mondiale* for *world war*. These candidate translations are displayed in the upper cells of the chart. Because the internal structure of noun phrases was not available for the translation process, we can translate a full noun phrase by decomposing it in several ways. For example *third world war* can be split up in the separate translation of either *third world* and *war* or in the

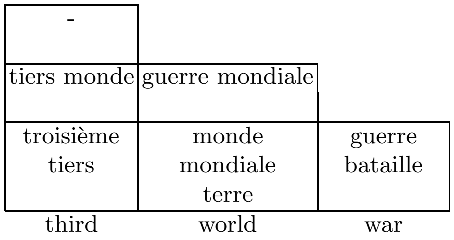

Fig. 4. translation chart of *third world war*

separate translation of *third* and *world war*. The most probable decomposition can be found using techniques developed for stochastic grammars (see e.g. [2]). The probabilities of the parse trees can be mapped into probabilities, or weights, of possible translations. A more detailed description of the algorithm can be found in [12].

4.5 Manual disambiguation

The manual disambiguation of the dictionary output was done by a qualified interpreter which also was a native speaker of English. She had access to the Dutch version of the topics and to the English dictionary output consisting of a number of possible translations per source language (Dutch) query word. For each Dutch word, one of the possible English translations had to be chosen, even if the correct translation was not one of them.

4.6 Other information

In the experiments described in this paper we ignored one important source of information: the multi-word entries in the Van Dale dictionaries. Multi-word expressions like for instance *world war* are explicitly listed in the dictionary. For the experiments described in this paper we only used word-by-word translations using the single word entries.

5 Experimental setup and results

In section 3 we identified three methods for query translation: using one translation per query term, using a unstructured query of all translations per source language query term and using a structured query of all translations per source language query term. Each method is assigned a number 1, 2 or 3. In section 4 five sources of information were identified that may be used by these methods: dictionary preference, pseudo frequencies, parallel corpora, context in noun phrases and human expertise. Given the five information sources we identified seven (two experiments were done both with and without normalisation) basic retrieval experiments or runs that are listed in table 1. Each experiment is labelled with a letter from *a* to *g*.

Table 1. disambiguation methods

run name	technique to weight translations / pick the best translation
run?a	no weights used / dictionary preferred translation.
run?b	weight by pseudo frequencies.
run?c	normalise weights of possible translations (run?a)
run?d	weight by normalised pseudo frequencies
run?e	normalised 'real' frequencies estimated from the parallel Agenda 21 corpus.
run?f	weight by using noun phrases from documents (including normalisation)
run?g	disambiguation by a human expert

The combinations of seven information sources and three methods define a total number of 21 possible experiments. After removing combinations that are redundant or not informative 15 experiments remain.

In the remainder of this section we will report the results of 15 experiments on the TREC cross-language task test collection [3] topics 1-24 (excluding the topics that were not judged at the time of TREC-6 leaving 21 topics). The Dutch topics were used to search the English documents. Experiments were compared by means of their non-interpolated average precision, average precision in short. Additionally, the result of each experiment will be compared with the result of a monolingual base line run, which is the result of queries based on the English version of the TREC topics. The monolingual run performs at an average precision of 0.403. All experiments were done with the linguistically motivated experimental retrieval engine developed at the University of Twente.

5.1 One translation runs

Table 2 list the results of the one translation runs. Normalisation of translation weights is not useful for picking the best translation. Therefore the table does not list **run1c** and **run1d**. (**run1d** would give exactly the same results as **run1b**.)

Table 2. results of 'one-translation' runs

run name	average precision	relative to baseline (%)
run1a	0.262	65
run1b	0.231	57
run1e	0.282	70
run1f	0.269	67
run1g	0.315	78

Not surprisingly, the manual disambiguated run outperforms the automatic runs, but it still performs at 78 % of the monolingual run. Translation ambiguity and missing terminology are the two primary sources of cross-language retrieval error [10]. We hypothesise that the loss of performance is due to missing terminology and possibly errors in the translation scripts. The 78 % performance of the monolingual base line is an upper bound on what is possible using a one-translation approach on the TREC cross-language collection.

The best automatic run is the run using corpus frequencies **run1e**. This is a surprise, because we used a relatively small corpus on the domain of the Twenty-One demonstrator which is *sustainable development*. Inspection of the topics however learns us that a lot of topics discuss international problems like air pollution, combating AIDS, etc. which fall directly in the domain of sustainable development.

The dictionary preferred run **run1a** performs reasonable well. The run using context from noun phrases **run1f** performs only a little better. Pseudo frequencies **run1b** are less useful in identifying the correct translation.

5.2 Unstructured query runs

Table 5.2 list the results of the unstructured query runs using all possible translations of each original query term. We experimented with all information sources except for the human expert.

A first important thing to notice is that the normalisation of the term weights is a prerequisite for good performance if all possible translations per source language query term are used in an unstructured query. Not

Table 3. results of 'unstructured query' runs

run name	average precision	relative to baseline (%)
run2a	0.180	45
run2b	0.162	40
run2c	0.268	67
run2d	0.308	76
run2e	0.305	76
run2f	0.275	68

using the normalisation, as done in **run2a** and **run2b** will drop performance to a disappointing 40 to 45 per cent of the monolingual base line.

More surprisingly, the pseudo frequency run **run2d** and the real frequency run **run2e** now perform equally well and both approach the upper bound on what is possible with the one translation approach (**run1g**). Although the pseudo frequencies are not very useful for identifying the best translation, they seem to be as realistic as real frequencies if used for weighting the possible translations.

5.3 Structured query runs

Table 4 lists the results of the structured query runs. Normalisation of term weights is implicit in the structured query, so **run3a** and **run3b** will give exactly the same results as **run3c** and **run3d** respectively.

Table 4. results of 'structured query' runs

run name	average precision	relative to baseline (%)
run3c	0.311	77
run3d	0.330	82
run3e	0.335	83
run3f	0.323	80

The four runs do not differ as much in performance as their unstructured equivalents, which suggests that the structured queries are more robust than the unstructured queries. Again, the pseudo frequency run **run2d** and the real frequency run **run2e** perform almost equally well. Three out of four runs perform better than the manually disambiguated 'one translation' run **run1g**.

6 Conclusion

This paper gives an overview of methods and information resources that can be used for cross-language information retrieval. Evaluation of these methods on the TREC cross-language collection indicates that using all possible translations for searching leads to better retrieval performance in terms of average precision than using just one (the best) translation.

In several early publications on cross-language retrieval [10, 11, 16] it was hypothesised that the document translation approach to cross-language retrieval leads to better retrieval performance than the query translation approach because there is more context available in documents for lexical disambiguation. Of course, lexical disambiguation is easier if there is more context available, but we claim that lexical disambiguation is not essential for good retrieval performance. In fact, table 4 shows that the best performing runs simply use all possible translations. The results of the manually disambiguated run suggest that not much can be gained by putting a lot of effort in explicit disambiguation of possible translations. If sophisticated search algorithms are used, disambiguation is done implicitly during searching. This suggests that the hypothesis that document translation leads to better retrieval performance than query translation might not be true after all: further research is needed on this topic.

The appendix of this paper describes some important steps in the development of new probabilistic retrieval models. It introduces a new method to rank documents using boolean structured queries and it introduces a new way to include statistical translation directly into statistical retrieval. In the cross-language retrieval experiments reported on here, boolean structured queries outperform the unstructured queries. In future publications we hope to show that this method, although it needs the Boolean queries to be in conjunctive normal form, is also useful in a monolingual setting with Boolean queries that are formulated directly by the user.

Acknowledgements

The work reported in this paper is funded in part by the Dutch Telematics Institute project DRUID. The reported TREC evaluations were funded in part by the EU projects Twenty-One (IE 2108) and Pop-Eye (LE 4234). The term weighting approaches and experiments presented in this paper were developed in co-operation with Wessel Kraaij from TNO-TPD Delft as a preparation for the official TREC experiments. He deserves the credits for the idea to add the manually disambiguated run to our experiments and for pointing out the reference to Harman's grouping method [4]. We are most grateful to Wessel for his advice and assistance. Furthermore, we like to thank Lynn Packwood for the manual disambiguation of the Van Dale dictionary output and Thijs Westerveld for implementing the interface on the corpus dictionary.

Appendix: probabilistic weighting algorithms

The weighting algorithms for structured and unstructured queries are based on the linguistically motivated probabilistic model of information retrieval [5, 7, 8]. This appendix gives an overview of the model and of its application to cross-language information retrieval.

A.1 An informal description of the underlying ideas

The linguistically motivated probabilistic model is based on advances made in the field of statistical natural language processing and uses probability theory in quite a different way than the classical probabilistic approaches to information retrieval, like e.g. the well-known Robertson / Sparck-Jones probabilistic model [18]. It uses a metaphor that is very similar to 'urn models' that are often used in introductory statistics courses [14]. Instead of drawing balls at random with replacement from an urn, we will consider the process of drawing words at random with replacement from a document. Suppose someone selects one document in the document collection; draws at random, one at a time, with replacement ten words from this document and hands those ten words (the query terms) over to the system. The system now can make an educated guess as from which document the words came from, by calculating for each document the probability that the ten words were sampled from it and by ranking the documents accordingly. The intuition behind it is that users have a reasonable idea of which terms are likely to occur in documents of interest and will choose query terms accordingly [17].

The model can be extended to Boolean queries by treating the sampling process as an AND-query and allowing that each draw is specified by a disjunction of more than one term. For example, the probability of first drawing the term *information* **and** then drawing either the term *retrieval* **or** the term *filtering* from a document can be calculated by the model introduced in this paper without any additional modeling assumptions.

Furthermore it can be extended with additional statistical processes to model differences between the vocabulary of the query and the vocabulary of the documents. Statistical translation can be added to the process of sampling terms from a document by assuming that the translation of a term does not depend on the document it was sampled from. Cross-language retrieval using e.g. Dutch queries on an English document collection uses the sampling metaphor as follows: first an English word is sampled from the document, and then this word is translated to Dutch with some probability that can be estimated from a parallel corpus.

A.2 Definition of the corresponding probability measures

Based on the ideas mentioned above, probability measures can be defined to rank the documents given a query. The probability that an unstructured query $T_1, T_2, \cdots, T_n$ of length n is sampled from a document with document identifier D is defined by equation 1.

$$P(T_1, T_2, \cdots, T_n|D) = \prod_{i=1}^{n} (\alpha_1 P(T_i) + \alpha_2 P(T_i|D)) \tag{1}$$

The probability measure is defined by a linear combination of global information $P(T)$ on the terms and local information $P(T|D)$ on the terms. The global information is added because some query terms do not occur even once in the document of interest. It is assumed that these terms are randomly selected from any of the documents in the entire collection. In section A.4 it is shown that this probability measure can be rewritten to a tf×idf term weighting algorithm. A somewhat similar probability function was used by Miller, Leek and Schwartz [13]. They showed that it can be interpreted as a two-state hidden Markov model in which α_1 and α_2 define the state transition probabilities and $P(T)$ and $P(T|D)$ define the emission probabilities.

The extension for Boolean queries as mentioned above is straightforward. For each draw, different terms are mutually exclusive. That is, if one term is drawn from a document, the probability of drawing e.g. both the term *information* and the term *retrieval* is 0. Following the axioms of probability theory (see e.g. Mood [14]) the probability of a disjunction of terms in one draw is the sum of the probabilities of drawing the single terms. Disjunction of m possible translations T_{ij} $(1 \leq j \leq m)$ of the source language query term on position i is defined as follows.

$$P(T_{i1} \cup T_{i2} \cup \cdots \cup T_{im}|D) = \sum_{j=1}^{m}(\alpha_1 P(T_{ij}) + \alpha_2 P(T_{ij}|D)) \qquad (2)$$

Following this line of reasoning, AND queries are interpreted similar as unstructured queries defined by equation 1. Or, to put it differently, unstructured queries are implicitly assumed to be AND queries.

Statistical translation is added to these probability measures by assuming that the translation of a term does not depend on the document it was drawn from. If $N_1, N_2, \cdots, N_n$ is a Dutch query of length n and a Dutch term on position i has m_i possible English translations T_{ij} $(1 \leq j \leq m_i)$, then the ranking as structured queries would be done according to equation 3

$$P(N_1, N_2, \cdots, N_n|D) = \prod_{i=1}^{n}\sum_{j=1}^{m_i} P(N_i|T_{ij})(\alpha_1 P(T_{ij}) + \alpha_2 P(T_{ij}|D)) \quad (3)$$

The translation probabilities $P(N_i|T_{ij})$ can be estimated from parallel corpora, or alternatively by any of the methods described in section 4. Equation 3 is the basis of the structured query runs **run3c-f** described in section 5.3. The experiments only differ in the way the translation probabilities are estimated, i.e. the disambiguation method that was used.

For the unstructured query runs **run2a-f** statistical translation was added by making the number of times a query term occurs in equation 1 proportional to the translation probabilities. For **run2a** and **run2b** translation frequencies instead of translation probabilities were used. The translation frequencies or probabilities can be multiplied with the query weights of table 5 (see section A.4). Again, the experiments only differ in the way the translation probabilities were estimated.

A.3 Parameter estimation

In information retrieval it is good practice to use the term frequency and document frequency as the main components of term weighting algorithms. Our probabilistic model does not make an exception. The term frequency $tf(t,d)$ is defined by the number of times the term t occurs in the document d. The document frequency $df(t)$ is defined by the number of documents in which the term t occurs. Estimation of $P(T)$ and $P(T|D)$ in equation 1, 2 and 3 was done as follows:

$$P(T_i = t_i) = \frac{df(t_i)}{\sum_t df(t)} \tag{4}$$

$$P(T_i = t_i|D = d) = \frac{tf(t_i, d)}{\sum_t tf(t, d)} \tag{5}$$

The value of the unknown parameter α_2 was determined by previous experiments on three different collections including the TREC cross-language collection [8]. For the experiments described in this paper we used $\alpha_2 = 0.15$. The value of α_1 is determined by the fact that $\alpha_1 + \alpha_2 = 1$.

A.4 Rewriting to presence weighting algorithm

Similar to the probabilistic model of Robertson and Sparck-Jones [18] probability measures for ranking documents can be rewritten into a format that is easy to implement. A presence weighting scheme (as opposed to a presence/absence weighting scheme) assigns a zero weight to terms that are not present in a document. Presence weighting schemes can be implemented using the vector product formula. This section presents the resulting algorithms. Rewriting equation 1 results in the formula displayed in table 5. It can be interpreted as a tf×idf weighting algorithm with document length normalisation as defined by Salton and Buckley [19].

vector product formula:	$\text{similarity}(Q, D) = \sum_{k=1}^{l} w_{qk} \cdot w_{dk}$
query term weight:	$w_{qk} = tf(k, q)$
document term weight:	$w_{dk} = \log(1 + \frac{tf(k, d)}{df(k) \sum_t tf(t, d)} \cdot \frac{\alpha_2 \sum_t df(t)}{\alpha_1})$

Fig. 5. tf×idf term weighting algorithm

If a structured query is used, the disjunction of possible translations as defined by equation 2 should be calculated first. As addition is associative, we do not have to calculate each probability separately before adding them. Instead, respectively the document frequencies and the term frequencies of the disjuncts can be added beforehand. The added frequencies can be used to replace $df(k)$ and $tf(k, d)$ in the weighting formula of table 5. The resulting ranking algorithm for Boolean queries was introduced earlier by Harman [4] for on-line stemming. Harman did not present her algorithm as an extension of Boolean searching, but instead called it 'grouping'. A somewhat similar approach for cross-language information retrieval was adopted by Ballesteros and Croft [1] by using a 'synonym operator' on term translations with more than one target term equivalent. The operator treats occurrences of all words within it as occurrences of a single pseudo term whose document frequency is the sum of *df*'s for each word in the operator.

If translation probabilities are available, the adding of respectively the document frequencies and the term frequencies of the disjuncts should be done as a weighted sum with the translation probabilities as weights.

References

1. L. Ballesteros and W.B. Croft. Resolving ambiguity for cross-language retrieval. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 64–71, 1998.
2. R. Bod. *Enriching Linguistics with Statistics: Performance Models for Natural Language*. Academische Pers, 1995.
3. M. Braschler, J. Krause, C. Peters and P. Schäuble. Cross-language information retrieval (clir) track overview. In *Procedings of the seventh Text Retrieval Conference (TREC-7)*, 1999.
4. D. Harman. How effective is suffixing? *Journal of the American Society for Information Science*, 42(1):7–15, 1991.
5. D. Hiemstra. A linguistically motivated probabilistic model of information retrieval. In C. Nicolaou and C. Stephanidis, editors, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL-2)*, pages 569–584, 1998.
6. D. Hiemstra. Multilingual domain modeling in Twenty-One: automatic creation of a bi-directional translation lexicon from a parallel corpus. In P.A. Coppen, H. van Halteren, and L. Teunissen, editors, *Proceedings of eightth CLIN meeting*, pages 41–58, 1998.
7. D. Hiemstra. A probabilistic justification for using tf.idf term weighting in information retrieval. *International Journal on Digital Libraries*, to appear.
8. D. Hiemstra and W. Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In *Proceedings of the seventh Text Retrieval Conference (TREC-7)*. NIST Special Publications, 1999.
9. D.A. Hull. Using structured queries for disambiguation in cross-language information retrieval. In *AAAI Symposium on Cross-Language Text and Speech Retrieval*. American Association for Artificial Intelligence, 1997.
10. D.A. Hull and G. Grefenstette. A dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, 1996.

11. W. Kraaij. Multilingual functionality in the Twenty-One project. In *AAAI Symposium on Cross-Language Text and Speech Retrieval*. American Association for Artificial Intelligence, 1997.
12. W. Kraaij and D. Hiemstra. Cross-language retrieval with the Twenty-One system. In E. Voorhees and D. Harman, editors, *Proceedings of the 6th Text Retrieval Conference TREC-6*, pages 753–761. NIST Special Publication 500-240, 1998.
13. D.R.H. Miller, T. Leek and R.M. Schwartz. BBN at TREC-7: using hidden markov models for information retrieval. In *Proceedings of the seventh Text Retrieval Conference, TREC-7*. NIST Special Publications, 1999.
14. A.M. Mood and F.A. Graybill, editors. *Introduction to the Theory of Statistics, Second edition*. McGraw-Hill, 1963.
15. D.W. Oard. A comparative study of query and document translation for cross-language information retrieval. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas (AMTA)*, 1998.
16. D.W. Oard and B.J. Dorr. A survey of multilingual text retrieval. Technical report, University of Maryland, 1996. http://www.ee.umd.edu/medlab/mlir/mlir.html
17. J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In W.B. Croft, A. Moffat, C.J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998.
18. S.E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.
19. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.
20. G. Salton and M.J. McGill, editors. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.