Twenty-One at TREC-7: Ad-hoc and Cross-language track

Djoerd Hiemstra
University of Twente, CTIT
P.O. Box 217, 7500 AE Enschede
The Netherlands
hiemstra@cs.utwente.nl
http://www.cs.utwente.nl/~hiemstra/

Wessel Kraaij
TNO-TPD
P.O. Box 155, 2600 AD Delft
The Netherlands
kraaij@tpd.tno.nl
http://www.tpd.tno.nl/TPD/smartsite304.html

Abstract

This paper describes the official runs of the Twenty-One group for TREC-7. The Twenty-One group participated in the ad-hoc and the cross-language track and made the following accomplishments: We developed a new weighting algorithm, which outperforms the popular Cornell version of BM25 on the ad-hoc collection. For the CLIR task we developed a fuzzy matching algorithm to recover from missing translations and spelling variants of proper names. Also for CLIR we investigated translation strategies that make extensive use of information from our dictionaries by identifying preferred translations, main translations and synonym translations, by defining weights of possible translations and by experimenting with probabilistic boolean matching strategies.

1 Introduction

Twenty-One is a 2 MECU project with 12 partners funded by the EU Telematics programme, sector Information Engineering. The project subtitle is "Development of a Multimedia Information Transaction and Dissemination Tool". Twenty-One started early 1996 and is currently in its evaluation phase. The TREC ad-hoc and CLIR tasks fit our needs to evaluate the system on the aspects of monolingual and cross-language retrieval performance. Partners in Twenty-One are: Getronics Software, TNO-TPD[1], DFKI GmbH, Xerox Research Center Europe, Highland Software, University of Twente, University of Tübingen, MOOI foundation, Environ, Climate Alliance, VODO and Friends of the Earth.

1.1 Cross-language retrieval in Twenty-One

The Twenty-One database consists of documents in different languages, initially Dutch, English, French and German but extensions to other European languages are envisaged. The primary approach to CLIR in Twenty-One is Document Translation (DT). There are certain advantages and disadvantages to DT:

- DT reduces the cross-language retrieval task to a monolingual search task
- The quality of a translation can in principle be better because the full document context is available. In the case of query translation there is often very little context.

[1]TNO participated also in the SDR and filtering tracks, cf. "TNO TREC7 site report: SDR and filtering", elsewhere in this volume

- DT is slow but can be done off-line, therefore the time constraints are less severe.
- DT requires a full translation of the document base for each supported language, which makes it not really scalable.

The DT approach in Twenty-One can be supplemented with or replaced by query translation depending an the application and collection size. A more elaborate description can be found in [7].

1.2 The Twenty-One retrieval system

The Twenty-One demonstrator system is on-line since March 1998.[2] It is based on two types of indexes:

1. a fuzzy phrase index based on n-gram search on phrases,
2. a standard Vector Space Model (VSM) index based on lemmas.

The first index type is well suited for short queries and interactive query refinement, whereas the VSM index is better suited for longer queries. Before a document is processed by the Twenty-One system it goes through a number of NLP preprocessing steps.

1. The language of the document is identified.
2. The document is translated to the other supported languages using the Logos machine translation system. A copy of the translated document is kept in the database to give a high quality preview of the document in the users preferred language.
3. The original document and the machine translated documents are tokenised, part-of-speech disambiguated and lemmatised using the morphological tools of Xerox. The lemmas are used to build the vector space index.
4. Part-of-speech tags are used to extract noun phrases from the documents using the TNO parser. Noun phrases are used to build the fuzzy phrase index.
5. For language pairs not supported by Logos, the lemmatised and noun phrase bracketed document is translated using the VLIS lexical database of Van Dale Lexicography. Translations are used to build the fuzzy index and the VSM index for the other languages. Again a copy is kept in the database for previewing in the user-preferred language.

1.3 Twenty-One in TREC

For the TREC-7 evaluations we did not use the Twenty-One system in the setup mentioned above, primarily because of the size of the CLIR document collection. Instead we followed a query translation approach. Because of its modular design we were able to build versions of the Twenty-One system that are specifically suited for the TREC tasks.

One of our main goals after the first TREC-participation in TREC-6 [8] was to upgrade the monolingual performance of our system to a level that is comparable with groups already participating in TREC. To test the baseline performance of our system we entered the ad-hoc task with a simple version of the Twenty-One system, only using the TNO VSM engine and Porters stemmer for English. For the CLIR task we were able to use most of the Twenty-One modules. We used

[2] http://twentyone.tpd.tno.nl/

the VSM engine for retrieval, but also the fuzzy index to recover from missing translations and spelling variants of proper names. As a preprocessing step before indexing and translation we used the Xerox morphological tools. The VLIS lexical database of Van Dale Lexicography was used for the translation of the queries.

This paper is organised as follows. In section 2 we describe our work on the ad-hoc task. Section 3 describes our work on the CLIR task. Section 4 will give concluding remarks.

2 The ad-hoc task

The main contribution of Twenty-One to the TREC ad-hoc task is an experiment with a new weighting algorithm. The weighting algorithm was developed from scratch within the linguistically motivated probabilistic model of information retrieval [2]. The model uses language models and techniques that find their origin in the field of statistical natural language processing, an approach that others are also beginning to investigate [11]. Advances already made in the field of statistical NLP (see e.g. [9] for an overview) are used to give a probabilistic justification for using *tf.idf* weights. The experiment described in this paper shows that the new weighting algorithm outperforms the Cornell version of BM25 algorithm on the TREC-7 collection.

2.1 A linguistically motivated probabilistic model of IR

In the linguistically motivated probabilistic model documents and queries are described by compound events. A compound event is an event that consists of two or more single events, as when a die is tossed twice or three cards are drawn one at a time from a deck [10]. The single events that define the compound event are the index terms in the collection. Given a document-id, index terms are assumed to be independent. The probability of the compound event should therefore be calculated by multiplying the probabilities of the single events as in equation 1.

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

The general idea is the following. Each document contains a small sample of natural language for which the retrieval system should build a statistical language model $P(T|D)$ where T is a single event. If the user enters a query $T_1, T_2, \cdots, T_n$ the system uses equation 1 to calculate the probability of that query given each possible value of the document-id D. Perhaps the most straightforward way to estimate the probabilities $P(T|D)$ would be maximum likelihood estimation. A maximum likelihood estimate maximises the probability of observed events and assigns zero probability to unseen events. This makes the maximum likelihood estimate unsuitable for directly estimating $P(T|D)$ because it would assign zero probability to each document that does not contain all of the query terms. The problem that many of the possible events do not occur in the actual data is a well known problem in the field of statistical NLP: the sparse data problem. There are a number of standard solutions to the sparse data problem, one of them being estimation by linear interpolation. It is possible to remove the zero probabilities by mixing the maximum likelihood model of $P(T|D)$ with a model that suffers less from sparseness like the marginal $P(T)$ as in equation 2.

$$P_{li}(T|D) = \alpha_1 P_{mle}(T) + \alpha_2 P_{mle}(T|D), \;\; 0 < \alpha_1, \alpha_2 < 1 \text{ and } \alpha_1 + \alpha_2 = 1 \tag{2}$$

In equation 2 global information $P(T{=}t)$ on the term t is mixed with local information $P(T{=}t|D)$ on the term. The mix of global and local information is determined by the value of α_1 (which also determines the value of α_2 by $\alpha_2 = 1 - \alpha_1$). It is standard practice in IR to use the document frequency for global information and the term frequency for local information. The document frequency $df(t)$ is defined by the number of documents in which the term t occurs. The term frequency $tf(t,d)$ is defined by the number of times the term t occurs in the document d. Equation 3 defines how probabilities are estimated from document frequencies and term frequencies.

$$P(T_i = t_i | D = d) = \alpha_1 \frac{df(t_i)}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i,d)}{\sum_t tf(t,d)} \tag{3}$$

2.2 Rewriting to vector product normal form

We used the TNO vector space retrieval engine for the TREC-7 experiments. Obviously the ranking formula based on equations 1 and 3 cannot be used directly in a vector space engine. There is however a ranking formula that produces the same ranking as the ranking formula introduced in the previous section which can be implemented using the vector product similarity measure. This can be shown by rewriting.

$$\begin{aligned} P(T_1 = t_1, \cdots, T_n = t_n | D = d) &= \prod_{i=1}^{n} (\alpha_1 \frac{df(t_i)}{\sum_t df(t)} + \alpha_2 \frac{tf(t_i,d)}{\sum_t tf(t,d)}) && (4) \\ &\propto \sum_{i=1}^{n} \log(1 + \frac{tf(t_i,d)}{df(t_i) \sum_t tf(t,d)} \cdot \frac{\alpha_2 \sum_t df(t)}{\alpha_1}) && (5) \end{aligned}$$

Equation 4 follows directly from equation 1 and 3. Multiplying equation 4 with values that are the same for each document, like α_1 and $df(t)$, will not affect the final ranking. Moreover, any monotonic transformation of the ranking formula will also produce the same ranking of documents. Instead of the product of weights we could therefore rank the documents by the sum of logarithmic weights. Using these two considerations the ranking formula can be rewritten as shown in equation 5. As a final step the query weights of the vector product formula can be used to account for multiple occurrences of the same term in the query. The resulting vector product version of equation 4 is displayed in table 1. Note that we end up with a document term weight that by the definition

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

Table 1: vector product version of weighting algorithm

of Salton et al. [13] can be interpreted as a *tf.idf* weight with document length normalisation. However, the $\sum_t tf(t,d)$ in the denominator of the document term weight in table 1 is the result of the requirement that probabilities have to sum up to one and not the results of document length normalisation. Document length normalisation *is* assumed by the fact that we ignore the prior probability $P(D)$. In fact we may assume that longer documents are more likely to be relevant by using the prior probability of equation 6 and ranking the documents using $P(T_1 = t_1, \cdots, T_n =$

$t_n|D = d) \cdot P(D = d)$.

$$P(D = d) = \frac{\sum_t tf(t, d)}{\sum_t \sum_d tf(t, d)} \quad (6)$$

This results in a weighting algorithm that cannot be rewritten into the vector product normal form. It can however be implemented fairly easily by initialising similarities to $\log(\sum_t tf(t, d))$ instead of to zero when processing the query. This version of the weighting algorithm was used in the TREC-7 experiments.

2.3 Pseudo relevance feedback

Supplementary runs were done with a Rocchio-like pseudo relevance feedback. After an initial retrieval run, the top 200 of the weighted index terms extracted from a concatenation of the top 3 documents were added to the query with a ratio of $20 : 3$, i.e. the weight of the added terms was multiplied by $3/20$ before adding. These parameters were determined empirically by experimenting with the English topics of the TREC-6 CLIR track.

2.4 Synonyms

We also experimented with query expansion based on synonyms taken from the Van Dale lexical database. The expansion worked well with the TREC-6 CLIR queries, but performed disappointing with the TREC-6 ad-hoc queries. No official run was submitted for this experiment

2.5 Experimental setup and results

For the ad-hoc task we built the simplest possible version of the Twenty-One system. We used the TNO vector space engine for indexing and Porters algorithm for stemming. Stopwords were removed from the documents, including words that are frequent in previous TREC topics like *relevant* and *document*. Queries were generated automatically from the full topics (title, description and narrative) using the same procedure as used for indexing.

As already mentioned above, the linguistically motivated weighting algorithm has one free parameter that defines the mix of local and global frequency information. We did three pilot experiments using Cranfield, the English CLIR collection topics 1-24 and the TREC-6 ad-hoc collection topics 251-300. The pilot experiments indicated that the best values for α_1 and α_2 are approximately the same for all three collections: $\alpha_1 = 0.85$ and $\alpha_2 = 0.15$. We used these values in the TREC-7 ad-hoc experiments.

runname	description
tno7cbm25	run using the Cornell version of BM25
tno7tw4	run using the linguistically motivated weighting algorithm
tno7exp1	run using the linguistically motivated algorithm with pseudo relevance feedback

Table 2: description of ad-hoc runs

According to Voorhees and Harman [16] the most popular weighting algorithm in TREC-6 was the Cornell implementation of the Okapi BM25 algorithm [12, 14]. We decided to compare the performance of the new weighting algorithm with the Cornell version of BM25, resulting in the first two runs listed in table 2. As a third run we submitted a run that uses the new weighting algorithm and pseudo relevance feedback.

runname	avg. prec.
tno7cbm25	0.2315
tno7tw4	0.2490
tno7exp1	0.2785

Table 3: results of ad-hoc runs

Table 3 lists the results of the three runs on the TREC-7 collection. The results show that the linguistically motivated weighting algorithm outperforms the Cornell version of BM25. The results also show a remarkable performance gain due to pseudo relevance feedback. The pseudo relevance feedback run performed above or equal to median on 44 of the 50 topics.

3 The CLIR task

In TREC-7 we intended to improve our TREC-6 results in the following ways:

- improve the monolingual system,
- improve lexical lookup,
- extend context sensitive disambiguation.

After reassessment of TREC-6 runs and experiments with manual disambiguation, we chose to experiment with an extension of the retrieval model in the direction of boolean interpretation. The manually disambiguated runs gave only a very small improvement in performance whereas initial experiments with a probabilistic interpretation of boolean structured translated queries (cf. sections 3.1 and 3.3) gave promising results on the TREC-6 CLIR topic set [3].

3.1 Query translation using the Van Dale dictionaries

The VLIS database from Van Dale Lexicography is a relational database which contains all lexical knowledge that is used for publishing the dictionaries Dutch → foreign language (German, French, English, Spanish). So the database is based on Dutch headwords with translation relations to equivalent lemmas in the foreign languages. The lexical material from the foreign language → Dutch companion dictionaries is not included in the VLIS database. Translation from one TREC language to another will go by using Dutch headwords that have the query words as their translations as interlingua, and then translating the Dutch headwords into one ore more words in the target language.

language	simple	composit	total
english	260k	40k	300k
german	224k	24k	248k
french	241k	23k	264k
spanish	139k	28k	167k

Table 4: number of translation relations in the VLIS database

The VLIS database contains simple and composite (multi-word) lemmas for 5 languages, Dutch being the pivot language. For Dutch there are 270k entries corresponding to about 513k concepts. These concepts have translations into French, Spanish, German and English. For TREC-7 we only used the simple lemmas. We used the the Xelda toolkit of Xerox Research Centre in Grenoble for tagging and lemmatisation of the topics. Translation of the topics was done in a series of steps:

1. Tokenising
2. Tagging
3. Lemmatisation
4. Stopword removal
5. Multi stage lexical lookup:
   - Lookup of compounds, subcompounds and compound parts
   - Lookup with and without syntactic constraint
   - Lookup of main / synonym translation
   - Lookup with or without capitalisation
   - Fuzzy lookup (not used in official runs)
6. Weighting of translations / Selection of best translation

The weighting of translations is based on the number of occurrences of a certain translation in the dictionary. Some head words carry over to the same translation for different senses. For example the Dutch head word *jeugd* can be translated to *youth* in three senses: the sense of 'characteristic', 'time-frame' and 'person'. The sense of 'person' has a synonym translation: *youngster*. As *youth* occurs in the dictionary under three senses we assign it a weight that is three times as high as the weight for *youngster*. Dutch serves as an interlingua, therefore translation can be carried out via several Dutch pivot lemmas. This possibly generates even more occurrences of the same possible translation. The implicit assumption made by weighting translations is that the number of occurrences generated from the dictionary may serve as rough estimates of actual frequencies in parallel corpora. Ideally, if the domain is limited and parallel corpora on the domain are available, weights should be estimated from actual data [4]. Weighting of possible translations is used for structured queries (see section 3.3).

A complication with respect to TREC-6 was the extension of the document base with Italian documents. We decided to use the machine translation system Systran[3] to handle our query translations to Italian. This Web service hosts translation capabilities from and to English for 5 European languages. For the German queries, we used English as a pivot language to translate to Italian because the language pair German-Italian is not available. Morphological stemming of the Italian translations were produced by ETH Zürich.

3.2 Fuzzy expansion of query terms

We developed a fuzzy expansion algorithm based on relative frequencies of letter trigrams and edit distance. This algorithm matches a query term with index terms that are similar but not exactly equal to the query term. This is useful for source language terms that do not have an entry in our dictionary, but do have a similar translation in the target language like domain specific jargon, person names and geographical names. It is also useful for spelling variation and spelling mistakes. We applied two forms of fuzzy expansion, a conservative version where only (translated) query

[3]http://babelfish.altavista.com

terms which are not found in the index are expanded. A more liberal version expanded *every* query term. The expansions were treated as a single concept by the TNO vector space engine. The fuzzy matching module ISM[4] was developed in the 1970's at TNO by de Heer [1].

3.3 Probabilistic interpretation of boolean queries

For almost every headword the VLIS lexical database gives a number of senses, each with a main translation and synonym translations. Instead of picking the preferred translation from the dictionary, it might be advantageous to use all possible translations to search for relevant documents as this might lead to higher recall. If possible translations are weighted and structured properly the document collection itself can be used for implicit disambiguation of possible translations [5]. Disjunction is a natural operator to combine possible translations of a query term, whereas conjunction is used to link all translations in a way that reflects the original query.

We developed a weighting algorithm that inputs boolean queries in conjunctive normal form and assigns probabilities to documents given these queries. Boolean queries are generated automatically from the topics by translation. The algorithm takes into account the relative frequencies of possible translations which are based on information from the VLIS lexical database as mentioned above. Details of the algorithm will be published in the near future.

3.4 Merging of runs

We created a separate index for each language. An alternative approach would be to build one big index on all four collections. The main reason for using four separate indexes is that our experimental setup was still based on experiments with only one target language. There is however a more fundamental reason for using four indexes. From the perspective of the linguistically motivated model of IR it would be silly to build a language model which mixes words of four different languages.

Merging the retrieval runs on separate collections into a combined run is not a trivial problem [6]. For the monolingual case, the problem is known as the *collection fusion problem* [15]: similarities are not comparable across collections because of the incorporation of collection-dependent frequency counts like document frequencies. One solution to the problem is to bias the similarities for each subcollection differently, e.g. by a collection specific linear transformation of the similarities.

We used the following approach to merging. Documents were retrieved from each collection and merged after adding a collection specific constant. The collection specific constant was determined by forcing the average similarities over all 28 topics to be the same for each language collection. The collection specific constant makes sure that, on average, the same number of documents is retrieved from each language. This approach is not an elegant solution to the merging problem. In fact, one could argue that it violates the TREC ad-hoc task description. We incorporated this method anyway to make sure that our contribution to the pool would not be biased towards one or two languages. In the near future we will develop a more elegant merging strategy.

3.5 Experimental setup and results

For the CLIR task we were able to use most of the Twenty-One modules. We used the VSM engine for retrieval and the fuzzy index to recover from missing translations and spelling variants of proper names. As a preprocessing step before indexing and translation we used the Xerox morphological tools. The translation of the topics was based on a word by word translation process, using the VLIS lexical database from Van Dale.

[4]ISM: Informatie Sporen Methode = Information Trace Method

runname	description
tno7edp	dictionary preferred translation of English queries into 3 other languages; fuzzy expansion of query terms without dictionary entry
tno7edpx	dictionary preferred translation of English query into 3 other languages; fuzzy expansion of each query term
tno7egr	probabilisticly interpreted boolean query of all possible translations of the English queries into 3 other languages
tno7ddp	dictionary preferred translation of German queries into 3 other languages; fuzzy expansion on query terms without dictionary entry
tno7eef	dictionary preferred translation of English query into French; fuzzy expansion of query terms without dictionary entry
tno7mx	unofficial run: merged run of four monolingual searches; fuzzy expansion of each query term

Table 5: description of CLIR runs

Table 6 lists the results for our official runs. As a baseline we included **tno7mx**, an unofficial run which is based on a merge of 4 monolingual runs. The best result is achieved by **tno7egr** the probabilistic boolean run. The more liberal fuzzy expansion run also produced improved results. It could be interesting to combine both techniques. The results with the German version of the topics is lower than expected, a topic-wise comparison is needed to assess the cause. **tno7eef**, the run with English topics in the English and French subcollection has a higher score, probably because of the reasons addressed in section 3.6 and because of the major contribution of the monolingual English run in the merged run.

runname	avg. prec.	relative to tno7mx (%)
tno7edp	0.2716	83
tno7edpx	0.2846	87
tno7egr	0.3009	92
tno7ddp	0.2382	73
tno7eef	0.3404	-
tno7mx	0.3282	100

Table 6: results of CLIR runs

Table 7 shows the results of our monolingual runs which were the building blocks for our merged runs. The table only shows the results on topics that had hits in all four languages: topics 26, 44, 46 and 51 are not included in this evaluation. For the per language evaluation we used the qrels from the merged runs. This will give a distorted picture (more about this aspect in section 3.6) but we think that the qrels are still useful to look at relative performances of runs for a particular language, and to get some idea of the merging effect. The merging effect turned out to be quite impredictable because the ranking of the runs based on column 6 is quite different from the ranking based on merged runs (column 7). For the average figure (column 6), the dictionary preferred runs are better than the probabilistic interpreted boolean run. However this ordering is reversed in the official ranking. Apparently the merging procedure that we applied is suboptimal. Further research is needed to explain these results.

runname	avg.prec. english	avg.prec. french	avg.prec. german	avg.prec. italian	average over 4	merged	relat. to avg. (%)
tno7edp	0.4923(m)	0.2893	0.2427	0.3846	0.3522	0.2750	78
tno7edpx	0.4985(m)	0.2945	0.2771	0.3927	0.3657	0.2901	79
tno7egr	0.4923(m)	0.3005	0.2253	0.3846	0.3506	0.3084	88
tno7ddp	0.3549	0.2452	0.4199(m)	0.3266	0.3367	0.2573	76
tno7mx(m)	0.4985	0.4542	0.4187	0.4651	0.4591	0.3569	78

Table 7: per language performance and the effect of merging on 24 topics TREC-7, (m) indicates monolingual run

3.6 Pool construction

The methodology for pool construction has some important but often neglected assumptions which complicate the evaluation of merging. The pool is based on a merge of the top 100 fraction of the runs which are judged. The assumption is that the pool will contain most relevant documents. Suppose that the average number of relevant documents per query is much larger than 100, then the validity of the assumption is questionable. In that case it is quite probable that a considerable amount of relevant documents is not judged because they are not in the pool. We computed some pool statistics, in particular we looked at the average judged fraction of the collection for each task. This is defined as the number of relevance judgements divided by the product of the number of queries and the collection size. The pool statistics are listed in table 8.

collection	total docs.	judged docs.	relevant docs.	no hits in topic	judged fraction	judged docs.	relevant docs.	no hits in	judged fraction
english	242,866	9,810	1,689	26,46	0.0014	8,713	1,385	8	0.0015
french	141,637	6,130	991	-	0.0015	12,663	1,518	22	0.0037
german	185,099	4,558	917	26	0.0009	9,086	1,172	22	0.0020
italian	62,359	3,062	501	26,44,51	0.0018	-	-	-	-
total	631,961	23,560	4,098	average:	0.0013	30,462	4,075	average:	0.0022

Table 8: CLIR task statistics (a) 28 topics TREC-7, (b) 24 topics TREC-6

The judged fraction of the ad-hoc task is 0.0030, which means that on average 3 per mill of the document collection has been judged for each query. For this year's CLIR track this figure is 0.0013 on average. This clearly reflects that the CLIR pool is much smaller than the ad-hoc pool. The pool is also smaller than the TREC-6 CLIR pool which has a judged fraction of 0.0022. Conclusion is that results derived from the CLIR pool are less reliable than last year. This might be partly due to the decision to base the pool on merged runs and not on monolingual runs.

The average number of relevant documents per topic is 93.5 in ad-hoc and 146.35 for the merged CLIR collection. For the monolingual subcollections the figures are: 60(EN), 35(FR), 33(DE) and 18(IT). This means that the "pool validity assumption" is probably violated for the merged CLIR task, meaning that the average precision figures for the merged runs are probably too high.

The figures for the monolingual results in table 7 are probably even more flattered. This can be explained as follows: suppose that we have subcollections of similar size, and that topics have roughly similar amounts of relevant documents in each subcollection. This means that on average only the top 25 documents of each subcollection run is judged in comparison to the top 100 documents for the merged runs. When we apply the qrels of the merged runs pool to the subcollection runs, the average precision is artificially high. Most probably a pool based on the top 100 of subcollection runs would bring in more relevant documents with lower ranks. This explanation also partly accounts for the high score of the merged run on the English French subcollections **tno7eef**.

4 Conclusion and outlook

The new linguistically motivated retrieval model outperforms the popular Cornell BM25 weighting scheme in the ad-hoc task with 7.5 %. Pseudo relevance feedback improves the average precision with an additional 12.7 %. The most successful run in the CLIR task was a probabilistic interpreted boolean run. However, at this point the merging process is ill understood. Evaluation of the merging process is complicated because the qrels of the merged runs cannot be used for a true comparison with subcollection runs. In future CLIR tracks, this problem should be tackled, for example by adding the top 100 documents from subcollection runs to the pool. There is some evidence that the 'probabilistic boolean' approach to CLIR is more successful than the 'dictionary preferred translation' approach. Further research is needed to assess whether 'boolean retrieval' is a better approach to the CLIR problem than disambiguation. A second result of the CLIR evaluation is that ISM (the fuzzy matching algorithm) gives a significant improvement in performance. The best result was obtained with the liberal variant: each query term is expanded with orthographically close variants, catching spelling variation in proper names and cognates in case of missing translations.

Acknowledgements

The work reported in this paper is funded in part by the Dutch Telematics Institute project Druid and the EU projects Twenty-One (IE 2108) and Pop-Eye (LE 4234). We would like to thank ETH Zürich for stemming the Italian collection, the Italian topics and our translations to Italian. Furthermore we would like to thank Renée Pohlmann and Rudie Ekkelenkamp both from TNO-TPD for their help with manual disambiguation of test runs and data preparation respectively.

References

[1] T. de Heer. Quasi comprehension on natural language simulated by means of information traces. *Information Processing & Management*, 15:89–98, 1979.

[2] 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.

[3] D. Hiemstra and F.M.G. de Jong. Cross-language retrieval in Twenty-One: using one, some or all possible translations? In *Proceedings of the 14th Twente Workshop on Language Technology (TWLT-14)*, pages 19–26, 1998.

[4] D. Hiemstra, F.M.G. de Jong and W. Kraaij. A domain specific lexicon acquisition tool for cross-language information retrieval. In *Proceedings of RIAO'97 Conference on Computer-Assisted Searching on the Internet*, pages 255–266, 1997.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] C. Manning and H. Schütze, editors. *Statistical NLP: Theory and Practice, draft*. http://www.sultry.arts.su.edu.au/manning/courses/statnlp/, 1998.

[10] A.M. Mood and F.A. Graybill, editors. *Introduction to the Theory of Statistics, Second edition*. McGraw-Hill, 1963.

[11] J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, 1998.

[12] S.E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 232–241, 1994.

[13] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[14] A. Singhal, G. Salton, M. Mitra and C. Buckley. Document length normalization. Technical Report TR95-1529, Cornell University, 1995. http://cs-tr.cs.cornell.edu/.

[15] E.M. Voorhees, N.K. Gupta and B. Johnson-Laird. The collection fusion problem. In D.K. Harman, editor, *Proceedings of the 3rd Text Retrieval Conference TREC-3*, pages 95–104. NIST Special Publication 500-225, 1995.

[16] E.M. Voorhees and D.K. Harman. Overview of the 6th text retrieval conference. In *Proceedings of the 6th Text Retrieval Conference TREC-6*, pages 1–24. NIST Special Publication 500-240, 1998.