Query Expansion by Local Context Analysis

Liana Ermakova* — **Josiane Mothe****

* *Institut de Recherche en Informatique de Toulouse, Perm State National Research University*
** *Institut de Recherche en Informatique de Toulouse*

RÉSUMÉ. *La tâche de notre recherche est de fournir le contexte de recherche à un moteur de recherche, i.e. étendre une requête. Nous proposons une méthode d'expansion automatique de requêtes basée sur la proximité des termes. Elle estime l'importance des termes candidats selon la proximité des termes de la requête dans les premiers documents retrouvés. L'évaluation sur les collections TREC indique que la performance de la recherche est améliorée significativement.*

ABSTRACT. *Query expansion (QE) aims at improving information retrieval (IR) effectiveness by enhancing the query formulation. Because users' queries are generally short and because of the language ambiguity, some information needs are difficult to answer. Query reformulation and QE methods have been developed to face this issue. Relevance feedback (RF) is one of the most popular QE techniques. In its manual version, the system uses the information on the relevance - manually judged- of retrieved documents in order to expand the initial query. Rather than using users' judgment on the document relevance, blind RF considers the first retrieved documents as relevant. Generally speaking, RF methods consider the terms that cooccur with query terms within positive feedback documents as candidates for the expansion. Rather than considering feedback documents in their all, it is possible to analyze local information. This paper presents a new method that uses local context from feedback documents for QE. The method uses POS information as well as the remoteness from query terms within feedback documents. We show that the method significantly improves precision on TREC collections.*

MOTS-CLÉS : *Recherche d'information, expansion automatique de requêtes, retour de pertinence, contexte local.*

KEYWORDS: *Information Retrieval, Query Expansion, Relevance Feedback, Local Context.*

1. Introduction

IR aims at retrieving the relevant documents according to a user's need. Concretely, a search engine computes a similarity between the user's query and the indexed documents ; the documents that contain the query terms are retrieved and ordered according to their decreasing similarity with the query. Because real queries are short and because natural language is ambiguous such matches can be wrong or incomplete. Matching is first difficult because the terms used by the authors of documents and by the users of search engines to represent a concept may be different. It is also difficult because users express their needs using just a few words, making the query difficult to "understand" by the system. The average query length remains between 2.4 and 2.7 words (Gabrilovich *et al.*, 2009). Various approaches have been developed to face these challenges. Semantic indexing and search aim at tackling the problem of term ambiguity. Some solutions rely on knowledge resources such as ontologies to use concepts rather than terms or stems, both during indexing and matching (Ermakova, 2015). Term ambiguity has also been treated with positive results as a classification or clustering problem, in which documents that share the same sense with the query terms are retrieved whereas documents that use the query terms but in a different meaning are filtered out (Schütze, 1998). On the other hand QE has driven many works in IR (see Carpineto's survey (Carpineto et Romano, 2012)). QE aims at adding new terms to the initial query based on some knowledge, either extracted from the term collection distribution, from the user or user's profile, or from feedback information. The initial query can be expanded using term cooccurrences in the documents (Amati, 2003 ; Xu et Croft, 1996) or based on Word-Net definition (Voorhees, 1994). The former type of approaches has the advantage of taking into account the document collection and thus the capability of the collection to contain the relevant information whereas the latter is collection independent. The use of relevance information was suggested by Rocchio (Rocchio, 1971) who defines the RF principle. Users are supposed to judge in some ways some of the retrieved documents and this feedback information is used in turn either to reweight query terms or to expand the query with the most important terms from relevant documents. To avoid users' judgment that can be difficult to collect and to make the process fully automatic, Buckley et al. (Buckley, 1995) suggested to consider the first initially retrieved documents as relevant. Many studies have shown that this method is efficient in general even if it can lower the performance for some queries (Amati *et al.*, 2004 ; Cronen-Townsend et Croft, 2002).

Selecting the most appropriate terms from the relevant -or considered as such- documents is a challenge. While weighting the term candidates considering their frequency or their weight calculated during the indexing phase is an intuitive and widely used approach, we suggest that a deeper analysis of document content can be useful. Searching for the best-fit terms in our approach is based on the local context analysis for ranking of terms and sentences. We use the term *local context analysis* for the analysis of neighborhood of a sentence or a term. Our first hypothesis is that terms that occur closely to query terms within the documents should be good candidates for QE ; the closer the better candidate. The second hypothesis is that natural language

considerations should help to decide the best candidate terms, that is to say that some types of terms should be better candidates (e.g. noun being better than adverbs). To study these hypotheses, we propose a method that considers a term window surrounding query terms from feedback documents. In addition, our method considers Part Of Speech (POS) information to weight differently the QE term candidates. The remainder of the paper is organized as follows. Section 2 presents related works. Section 3 details the QE method we promote. Section 4 presents the experimental framework as well as the collections and performance measures we used. Section 5 provides the results. Finally, section 6 concludes the paper and draws up some future works.

2. Literature Review

Automatic methods for QE were firstly proposed by Maron and Kuhns in 1960. QE based on RF makes the hypothesis that relevant documents are key components to decide which terms are important to formulate an enhanced query regardless to an information need. Using the vector space model, Rocchio defined a way to re-weight query terms and thus to add new terms to the initial query - terms that were initially associated with a null value. The term weights are re-computed so that the terms that occur in relevant documents contribute positively to the new query whereas the weight of the terms that occur in non-relevant documents are lowered (Rocchio, 1971). A balance between the initial query and feedback information is involved in the weighting. While improving the effectiveness of search, the method however implies that document relevance is collected. Buckley et al. (Buckley, 1995) went a step further by assuming the top-retrieved documents are relevant. The so-called blind or pseudo RF is now commonly used in IR evaluation campaigns. Some current studies focus on the when QE is useful. Indeed it has been shown that if RF successfully improves the system performance in average (Voorhees et Harman, 1998), in some cases, QE worsens the quality of the retrieval (Amati *et al.*, 2004). Some approaches considers selective QE : the system decides whether or not QE should applied, based on some query features (Cronen-Townsend et Croft, 2002). The query features are either pre-retrieval or post-retrieval query features that are used to cluster queries. A training phase builds the model from queries for which the best decision is know ; then the model is applied to any new query. Another range of works focuses on selecting the best feedback information. Cao et al. propose a term classification method to predict the usefulness of expansion term candidates (Cao *et al.*, 2008). Xu et al. (Xu *et al.*, 2009) suggested that top documents should not be considered in a blind way but rather the system should distinguish relevant from non-relevant documents. Their idea is that non relevant documents should cluster as relevant documents do. In addition, they consider that query terms should occur in the relevant document cluster and that some documents from the non-relevant cluster do not contain any of the query terms. One can consider term cooccurrence by applying latent semantic analysis hypothesizing that related words cooccur in similar context (Landauer *et al.*, 1998). Term co-occurrence may be discovered by a cluster algorithm, e.g. the Naive-Bayes maximizing the classification maximum likelihood criterion, where each word is pre-

sented as a vector with the components corresponding to the number of occurrences of the word in each document (Amini *et al.*, 2007). Singh and Sharan combined co-occurrence and semantic similarity of terms (Singh et Sharan, 2015). Xu and Croft went a step further (Xu et Croft, 1996). They use a feature selection based on co-occurrence of terms, considering that the best terms are the ones that co-occur with as many query terms as possible within the top-ranked documents or document passages. In addition, they consider nouns and noun phrases as the expansion terms. Xu and Croft's co-occurrence measure is not a probability in the strict sense, while mutual information shows the joint probability of terms to co-occur within a text. Wan *et al.* suggested to combine ontology-based methods with the proximity heuristics (Wan *et al.*, 2012). Miao *et al.* proposed an extension of the Rocchio's approach by introducing a concept of proximity-based term frequency that focuses on the proximity of terms rather than positional information unlike the positional relevance model (Miao *et al.*, 2012). They provide 3 approaches to estimate the proximity-based term frequency, namely (1) moving window ; (2) kernel-based and (3) Hyperspace Analogue to Language (HAL) methods. In contrast to (Wan *et al.*, 2012 ; Miao *et al.*, 2012), we estimate the distance in term of sentences and we evaluate the sentences that are the sources of the candidate terms.

The method we propose also considers feedback information for QE. We consider the co-occurrence of query terms with QE term candidates from pseudo-relevant documents. Moreover, we make use of the surrounding terms of query terms. Our hypothesis is that the closer a term to a query term, the better as a QE term candidate. POS information is used in order to weight differently QE candidates.

3. Method Description

In this paper a *token* is viewed as a word occurrence within a document. Moreover, we will use *token* and *word* as synonyms. A term is a normalized representation of tokens in a dictionary (Manning *et al.*, 2008) (i.e. stems or lemmas). The key idea of the proposed method is to search the most appropriate terms for QE in the top ranked documents. Searching for the best-fit terms is based on ranking of terms and sentences. Both ranking procedures include local context analysis, i.e. analysis of neighboring sentences. Our approach is underlain by the following hypotheses :

1) Not always an entire document is relevant to a query, but it can contain one or several relevant passages. Term candidates should be selected from these passages.

2) Terms for QE come from appropriate sentences (in general, this hypothesis is similar to those of RF). The measure of sentence appropriateness is called sentence score and referred to $score(S)$ in the rest of the paper.

3) Good terms should have appropriate POS and high IDF. Not all POS are suitable for QE (e.g. functional words). Moreover, the most frequent terms are nouns. However, in some cases adjectives, verbs and numbers are indispensable. A good term should well distinguish documents from each other. POS weight and IDF may be

considered as a query-independent term score.

4) The terms lying in the neighborhood of query terms are closer related to them than the remote ones.

Thus, all candidate terms are ranked according to the following metric :

$$w_{total}(t) = f(score(S), w_{pos}(t), IDF(t), importance(t, Q)) \quad [1]$$

where $score(S)$ is score of the sentence S containing t, $w_{pos}(t)$ is the weight of the POS of t, $IDF(t)$ is the inverse document frequency of the candidate term, $importance(t, Q)$ is a function of (1) the distance to the query Q terms, (2) their weights, and (3) the likelihood of the candidate term to cooccur not by chance with the query terms in the top ranked documents. $importance(t, Q)$ allows to find terms occurring in the neighborhood of important query terms. These elements are described in the following sections.

3.1. *Sentence Scoring*

Sentence scoring method is an elaboration of RF. We strengthen the criteria of provenance of good terms for QE used in RF. Sentence scoring method presented in this paper is the adaptation of the method initially developed for query-biased multi-document summarization (Ermakova, 2015). We assume that appropriate terms come from appropriate sentences. Right sentences should be of high quality and moreover they should match the query. Hence, sentence score is estimated as the function of sentence weight $ws(S)$ and sentence quality measure $SntQual(S)$ which is used to avoid trash passages from real web collections :

$$score(S) = \sigma(ws(S), SntQual(S)) \quad [2]$$

We define it as the function of the lexical diversity $LexDiv(S)$, meaningful word ratio $Meaning(S)$ and punctuation score $PunctScore(S)$:

$$SntQual(S) = \phi(LexDiv(S), Meaning(S), PunctScore(S)) \quad [3]$$

Lexical diversity allows avoiding sentences that do not contain terms except those from a query. Lexical diversity in our approach is defined as the number of different lemmas used within a sentence divided by the total number of tokens in this sentence. **Meaningful word ratio** is also aimed to penalized sentences that either have no sense at all or are not comprehensible without large context. Meaningful word ratio is the number of non-stop words within a sentence over the total number of tokens in this sentence. Besides unreadable passages, many symbols usually used as punctuation marks can be found in emoticons. Emoticons represents humans' attitude towards something. However, they are not relevant for informative, navigational nor transactional queries. Hence, $PunctScore(S)$ penalizes sentences containing many punctuation marks. **Punctuation score** is estimated by the formula :

$$PunctScore(S) = 1 - \frac{PunctuationMarkCount(S)}{TokenCount(S)} \quad [4]$$

where $PunctuationMarkCount(S)$ is a total number of punctuation marks in the sentence, and $TokenCount(S)$ – is a total number of tokens in S. Thus, $PunctScore(S)$ shows the ratio of tokens which are not punctuation marks.

Thus, we believe that a good sentence should have high ratio of different meaningful words and reasonable ratio of punctuation. Sentence quality is query-independent, while sentence weight $ws(S)$ shows how well a sentence matches a query.

We assume that relevant sentences come from relevant documents. However, in real world search engines we do not know which documents are actually relevant. Therefore in our method sentence weight depends on pseudo-relevance $DocRel(d)$ of the corresponding document d assigned by a search engine (i.e. document rank, score or their combination), and computed smoothed sentence relevance $R(S)$:

$$ws(S) = \omega(DocRel(d), R(S)) \quad [5]$$

Smoothed sentence relevance estimation $R(S)$ in (5) is based on the first-stage local context analysis. In contrast to (Yang *et al.*, 2011), we believe that a context does not provide redundant information, but allows to precise and extend sentence meaning. Neighboring sentences influence the sentence of interest, but this influence decreases as the remoteness of the context increases. We choose the simplest dependence model, namely the linear function. In this case $R(S)$ is calculated by the formulas :

$$R(S) = \sum_{i=-k}^{k} w_i \times r_i \quad [6]$$

$$w_i = \begin{cases} \frac{1-w_t}{k+1} \times \frac{k-|i|}{k} & 0 < |i| \leq k \\ w(S), & i = 0 \\ 0, & |i| > k \end{cases} \quad [7]$$

$$\sum_{i=-k}^{k} w_i = 1 \quad [8]$$

where $w(S)$ is the weight of the sentence S, w_i and r_i are respectively the weights and the prior scores of the sentences from the context of S of k length. In this formula weights of context linearly decrease with the growth of the distance from the sentence S. If the sentence number in left or right context is less than k, their weights are added to the target sentence weight $w(S)$. This allows keeping the sum equal to one. That is important since otherwise a sentence with a small number of neighbors (e.g. the first or last sentences) would be penalized (even if first sentences of a document are often considered to be very informative). Prior scores of sentence r_i correspond to their prior similarity to the query $sim_{total}(S, Q)$ which is a function of the cosine similarity measure sim_{uni} between the terms of the sentence and the query and the similarity of named entities sim_{NE} :

$$r_i = sim_{total}(S, Q) = \zeta(sim_{uni}, sim_{NE}) \quad [9]$$

3.2. *Term Weighting*

The next step of our method is to compute the importance of all terms in all sentences from RF :

$$importance(t, Q) = \theta(wd(t, Q), coocurrence(t, Q)) \quad [10]$$

$wd(t, Q)$ is a function of the distance from the candidate terms to the query Q and their weights, and $coocurrence(t, Q)$ shows the likelihood of the candidate term to occur not by chance with the query terms in the top documents ranked according to the initial query. The concrete functions are given in the evaluation section.

The term score is combined with the corresponding sentence score. Thus, we used a two-step local context analysis : for sentence scoring and for estimation of term importance. In previous works local context was viewed as a single document and it was opposed to the entire collection analysis (global context) (Carpineto et Romano, 2012 ; Xu et Croft, 2000). In this paper we consider local context in a stricter way, precisely we look not only to the whole document statistics, but also for terms surrounding the query terms.

4. Experimental Framework

In this section the experimental framework is described. Firstly, we present the details of the implemented system. Then we provide the data sets we used and the evaluation metrics. The last subsection describes the systems used for comparison.

4.1. *Details of the Implemented System*

Our approach requires RF. In order to obtain preliminary ranking we used the Terrier platform[1], an open-source search engine developed by the School of Computing Science, University of Glasgow.This platform considers documents as bags of words. It implements various weighting and retrieval models. We applied a default retrieval model in Terrier $InL2$. $InL2$ is a DFR (Divergence from Randomness) model based on $TF - IDF$ measure with $L2$ term frequency normalization (Amati, 2003). The DFR models are based on the assumption that informative words are relatively more frequent in relevant documents than in others (Amati, 2003). In the DFR models the weight $weight(t, d)$ of the term t in the document d is estimated as follows :

$$weight(t, d) = \frac{1}{tf + 1} \times Inf_1(tf) \quad [11]$$

$$Inf_1(tf) = tf \times \log_2 \frac{N + 1}{n + 0.5} \quad [12]$$

1. terrier.org

where tf is the initial frequency of the term t in the document d, N is the total number of documents in the collection, and n is the number of documents containing the term t. $L2$ normalization of the term frequency tf_n is computed as :

$$tf_n = tf \times \log_2 1 + c\frac{avg_l}{l},\ c > 0 \quad [13]$$

where l is the length of the document d, avg_l is the average document length, and c is the normalization parameter. Thus, $weight(t, d)$ determined by $InL2$ is :

$$weight(t, d) = \frac{tf_n}{tf_n + 1} \times \log_2 \frac{N + 1}{n + 0.5} \quad [14]$$

Stemming was performed by Porter's algorithm. The 10 top documents retrieved for each query were processed by Stanford CoreNLP which integrates POS tagger based on the Penn Treebank tag set and named entity recognizer[2]. The parsed documents were treated as described in the section 3.

For the parameter combination one can use various functions : sum, weighted sum, product, exponential function or more complex non-linear functions. Exponential function may be reduced to product. In our experiments we mostly used product function to combine parameters. It was caused by several reasons. Firstly, the product of parameters is biased to the smallest one. It allows penalizing candidates that are not suitable according to one of the score elements, e.g. the sums of $0.9 + 0.1 = 1$ and $0.5 + 0.5 = 1$ are equal, but the products $0.9 \times 0.1 = 0.09$ and $0.5 \times 0.5 = 0.25$ differ much. The same refers to weighted sum. The only difference is the impact of each parameter in the result. Moreover, multiplication does not require normalization. In our future work we will try more complex functions to combine parameters. Hence, the total term score was estimated as the product of the score of the sentence S containing the term t, its POS weight, IDF and the likelihood of t to cooccur not by chance with the query terms :

$$w_{total}(t) = score(S) \times w_{POS}(t) \times IDF(t) \times importance(t, Q) \quad [15]$$

The weights $w_{POS}(t)$ of different POS were set as parameters.

Sentence score was computed as the product of the sentence weight $ws(S)$ and its quality $SntQual(S)$:

$$score(S) = ws(S) \times SntQual(S) \quad [16]$$

Sentence weight $ws(S)$ was calculated by the formula :

$$ws(S) = DocRel(d) \times R(S) \quad [17]$$

Sentence quality measure $SntQual(S)$ was estimated as the product of of the lexical diversity $LexDiv(S)$, meaningful word ratio $Meaning(S)$ and punctuation score $PunctScore(S)$:

$$SntQual(S) = LexDiv(S) \times Meaning(S) \times PunctScore(S) \quad [18]$$

2. http ://stanfordnlp.github.io/CoreNLP/

The similarity between a sentence and a query was computed as follows :

$$sim_{total}(S,Q) = sim_{uni} \times sim_{NE} \quad [19]$$

$$sim_{NE} = \frac{NE_{common} + NE_{weight}}{NE_{query} + 1} \quad [20]$$

where NE_{weight} is positive floating point parameter, NE_{common} is the number of NE appearing in both query and sentence, NE_{query} is the number of NE appearing in the query. NE_{weight} allows not to reject sentence without NE which can be still relevant. We add 1 to the denominator to avoid division by zero.

$$importance(t,Q) = wd(t,Q) \times cooccurrence(t,Q) \quad [21]$$

$$wd(t,Q) = \max_{Q_i}(IDF(Q_i) \times d(t,Q_i)) \quad [22]$$

$$d(t,Q_i) = \frac{1}{\max(1, dist(t,Q_i) - 2)} \quad [23]$$

where $dist(t,Q_i)$ is the distance from the term to the nearest term appearing in the query. We believe that terms cooccurring in the window of the length 4 are strongly interconnected, thus we consider them to have the same weight by applying the formula $\max(1, dist(t,Q_i) - 2)$. Here we use the same hypothesis as for sentence scoring, namely the interconnection between text parts decreases as the distance between them grows. The likelihood of the candidate term to occur not by chance with the query terms in the top documents ranked according to the initial query can be calculated by different models (see the detailed description of Xu and Croft's approach and DFR models in subsection 4.4). In our approach it was estimated similarly to Bose-Einstein 2 model (Amati, 2003). We consider only terms occurring at least in 2 top ranked documents. In contrast to (Amati, 2003), we apply grammatical filters to the candidate terms, i.e. we consider only some POS (e.g. only nouns or all meaningful words), while others are ignored (e.g. numbers, pronouns, adverbs etc.). Another modification is that we filter words before scoring.

4.2. *Data Sets*

The evaluation was performed on two kinds of datasets : TREC Ad Hoc Track data sets and WT10G [3]. Documents are tagged by SGML. The collected documents are not normalized and may contain spelling or other errors. TREC Ad Hoc Track data sets are "pure" collections since the documents have almost the same format and there is no spam. In contrast, WT10G is a snapshot of the web with real documents in HTML format, some of which are spam.

3. http ://trec.nist.gov/

We used TREC Ad Hoc Track data sets (Text Retrieval Conference) for three years (TREC) : 1997 (6), 1998 (7) and 1999 (8). TREC 6-8 are driven on the data on Disks 4 and 5 and contain 150 topics in total. There are 4 sources of documents : the news articles from (1) The Financial Times, 1991-1994 (FT) - 564MB, 210 158 documents ; (2) Federal Register, 1994 (FR94) - 395MB, 55 630 documents ; (3) Foreign Broadcast Information Service (FBIS) - 470MB, 130 471 documents ; and (4) The LA Times - 475MB, 131 896 documents. Each of TREC 6-8 has 50 topics. A topic represents an information need and contains 4 fields : (1) topic number, (2) title (very short description of a topic – about three words), (3) description (a "normal" sentence description of a topic), and (4) narrative (description of the information that should be presented at relevant documents). The pools of relevant documents (q-rels) were merged from the top 100 documents per topic retrieved in each submitted run and assessed by humans. In total for 150 queries 14 013 documents were considered to be relevant.

WT10G was used at TREC Web track 2000-2001. It is 10GB subset of the web snapshot of 1997 from Internet Archive. WT10G contains 1 692 096 documents from 11 680 servers (minimum 5 documents per server). There were 50 topics in 2000 and 2001 (total 100 topics). In total 5 953 were judged as relevant.

4.3. *Evaluation Measures*

TREC q-rels may be used by trec_eval. The trec_eval software enables to evaluate ranked retrieval results and implements the following measures :

– Mean Average Precision over all queries ;

– R-Precision ;

– Normalized discounted cumulative gain ;

– Precision at 5, 10, 15, 20, 30, 50, 100, 200, 500 and 1000 ;

– 11-point interpolated average precision (at 0%, 10%, ..., 100% of recall) which allows to draw precision-recall curve.

Precision (P) is the fraction of retrieved documents that are relevant (Manning *et al.*, 2008) :

$$P = \frac{\#RelevantRetrievedItems}{\#RetrievedItems} \quad [24]$$

Precision at k ($P@k$) the fraction of the top k retrieved documents that are relevant. Interpolated average precision is the ration of relevant retrieved documents over the number of documents that gives a certain percentage of recall. Recall (R) shows the fraction of relevant retrieved documents over all relevant documents :

$$R = \frac{\#RelevantRetrievedItems}{\#RelevantItems} \quad [25]$$

R-Precision is the proportion of relevant retrieved documents over the size of the set of relevant documents.

Mean average precision is calculated as follows :

$$MAP = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk}) \quad [26]$$

where $|Q|$ is the number of queries, m_j is the number of relevant documents for the j-th query, and R_{jk} is the set of ranked retrieval results from the top result until the document d_k is achieved.

Discounted cumulative gain DCG is a measure of IR effectiveness that penalizes highly relevant documents appearing lower in a search result (Manning *et al.*, 2008). The graded relevance value is discounted logarithmically proportional to the position of the result :

$$DCG_k(Q_j) = \sum_{i=1}^{k} \frac{2^{rel_i^{(j)}} - 1}{log_2(i+1)} \quad [27]$$

Normalized discounted cumulative gain $NDCG$ is normalized over Ideal DCG $IDCG$, i.e. the maximum possible DCG till the position k :

$$NDCG_k(Q_j) = \frac{DCG_k(Q_j)}{IDCG_k(Q_j)} \quad [28]$$

$NDCG$ can be averaged over all queries and all positions :

$$NDCG(k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} NDCG_k(Q_j) \quad [29]$$

$$NDCG = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} NDCG_k(Q_j) \quad [30]$$

4.4. *Systems Used for Comparison*

For evaluation purpose we used several RF methods, namely Xu and Croft's cooccurence model and DFR models implemented in Terrier. During QE the best-scored terms from the top-ranked documents are extracted. Terms are ranked using one of the DFR weighting model.

In the **Kullback-Leibler** model QE is performed by ordering the candidate terms by their information content given the query Q (Amati, 2003) :

$$Inf(t \mid Q) = Inf_{D_Q}(t) = -\log P(t \mid Q) \quad [31]$$

where t is the candidate term. If $P(t \mid Q)$ is viewed as binomial distribution :

$$\begin{aligned} P(t \mid Q) &= B(C_{D'(t)}, C_{D'}, \frac{C_D(t)}{C_D}) \\ &= \binom{C_{D'}}{C_{D'}(t)} (\frac{C_D(t)}{C_D})^{C_{D'}(t)} (1 - \frac{C_D(t)}{C_D})^{C_{D'} - C_{D'}(t)} \end{aligned} \tag{32}$$

where D' is a subset of the original collection D, $C_{X(t)}$ is the number of time the term t occurs in X, C_X – the total number of terms in X. In the case of the approximation via the divergence function the information content of the term t is proportional to :

$$Inf(t \mid Q) \sim TF_{D'}(t) \times \log \frac{TF_{D'}(t)}{TF_D(t)} \tag{33}$$

The information content of the term t in the **Chi-square** model is estimated as (Amati, 2003) :

$$\begin{aligned} Inf(t \mid Q) \sim TF_{D'}(t) \times TF_{D'} \times (\log \frac{TF_{D'}(t)}{TF_D(t)} + \log \frac{1 - TF_{D'}(t)}{1 - TF_D(t)}) \\ + 0.5 \times (2\pi \times TF_{D'} \times (1 - \frac{TF_{D'}(t)}{TF_D(t)})) \end{aligned} \tag{34}$$

Bose-Einstein 1 (Bo1) and **2 (Bo2)** models are DFR models implemented in Terrier (Amati, 2003). By default they are parameter-free, but Rocchio's QE mechanism can be also applied.

$$Bo1 = TF_{D'}(t) \times \log \frac{1 + f1}{f1} + \log (1 + f1) \tag{35}$$

$$f1 = \frac{TF_D(t)}{|D|} \tag{36}$$

$$Bo2 = TF_{D'}(t) \times \log \frac{1 + f2}{f2} + \log (1 + f2) \tag{37}$$

$$f2 = \frac{TF_{D'}(t) \times TF_{D'}}{TF_D} \tag{38}$$

We implemented the **Xu and Croft's** method described in (Xu et Croft, 2000) both for the entire documents and documents split at blocks of 300 words. The Porter's algorithm was used for stemming. The cooccurrence $coocurrence(t, Q)$ measure shows the likelihood of a term to occur not by chance with the query terms in the top ranked document set. $coocurrence(t, Q)$ is estimated as follows :

$$coocurrence(t, Q) = \prod_{Q_i} (\delta + co_{degree}(t, Q_i))^{IDF(Q_i)} \tag{39}$$

$$co_{degree}(t, Q_i) = \log(\sum_{d \in D'} TF(t, d) \times TF(Q_i, d) + 1) \times \frac{IDF(t)}{\log n} \quad [40]$$

The parameter δ was set to 0.05. IDF was estimated as :

$$IDF(t) = \min(1.0, \frac{\log_1 0 \frac{|D|}{|D'|}}{5}) \quad [41]$$

where D' is the set of the top-ranked documents from the collection D.

5. Results

We compared our system (LC) with those described in the previous section (see tables 1 and 2), namely :

– Kullback-Leibler divergence model implemented in Terrier (KL) ;

– Chi-square divergence model from Terrier (CS) ;

– Bose-Einstein 1 model from Terrier (Bo1) ;

– Bose-Einstein 2 model from Terrier (Bo2) ;

– Baseline presented by InL2 model without any QE (InL2) ;

– Xu and Croft's approach.

All systems used InL2 model for RF, 3 documents from which 10 best scored terms were extracted. The results obtained by applying Xu and Croft's approach were much lower than the baseline. Therefore, we do not present them.

Table 1 provides information about the number of relevant retrieved documents (RRD), MAP, R-precision (R-P) and NDCG. MAP and R-precision may be viewed as the main measures since MAP has very good discrimination and stability and R-precision represents the break-even point and highly correlates with MAP (Manning *et al.*, 2008). On both data sets by all metrics our systems showed the best results, which are much higher than the baseline. Table 2 contains the values of precision at k for the systems mentioned above. In most cases users prefer to look for relevant documents on the first page of search results. The standard size of search engine result pages is 10. Therefore, precision at 5 and 10 seems to be very important. At this level our system showed the best results for both data sets as well as for P@100. Table 3 provides paired difference Student t-test results performed on average precisions. Our method had the following characteristics Mean = 0.253620, Std.Dv. = 0.217502 and Mean = 0.220422, Std.Dv. = 0.20096 at TREC 6-8 and Web data sets respectively. At the level $p < 0.05$ the differences between the approach proposed in this paper and all other evaluated methods are significant for TREC 6-8 data. On Web data at the level $p < 0.05$ the differences between the our approach and Bo2 is insignificant, as well as the differences between LC and KL. The insignificant difference between LC and Bo2 can be caused by the fact that the design of our system which implies modified Bo2-score computation.

Tableau 1. *General results*

		LC	Bo2	KL	Bo1	CS	InL2
TREC 6-8	RRD	**8242**	8184	8034	8027	7913	7225
	MAP	**0.2536**	0.2491	0.2400	0.2391	0.2312	0.2105
	R-P	**0.291**	0.2868	0.2797	0.2818	0.2611	0.2627
	NDCG	**0.5221**	0.5168	0.5093	0.5087	0.4921	0.4682
WT10G	RRD	**3964**	3935	3938	3970	3724	3810
	MAP	**0.2204**	0.219	0.212	0.2094	0.2068	0.1894
	R-P	**0.2485**	0.241	0.2375	0.2355	0.242	0.2304
	NDCG	**0.4844**	0.4795	0.4816	0.4816	0.4515	0.4624

Tableau 2. *Precision at k*

		LC	Bo2	KL	Bo1	CS	InL2
TREC 6-8	P@5	0.496	0.4933	0.4800	0.4773	0.4547	0.4413
	P@10	0.4427	0.4413	0.4333	0.4327	0.4107	0.4180
	P@100	0.2235	0.2190	0.2133	0.2119	0.2048	0.1953
	P@1000	0.0549	0.0546	0.0536	0.0535	0.0528	0.0482
WT10G	P@5	0.3816	0.3796	0.3714	0.3694	0.3551	0.3286
	P@10	0.3378	0.3296	0.3255	0.3163	0.3102	0.2816
	P@100	0.1599	0.1589	0.1523	0.1510	0.1480	0.1387
	P@1000	0.0404	0.0402	0.0402	0.0405	0.0380	0.0389

6. Conclusion

In this paper we proposed the method for QE based on cooccurrence measure as well as importance estimated by analyzing local context. In contrast to previous works we treated not only entire documents, but also text passages surrounding query terms. We evaluated our method on TREC Ad Hoc 6-8 collection as well as on WT10g. Our approach obtained the best results among 6 systems, namely the baseline and DFR models for QE implemented in Terrier platform. Our system outperformed others according to MAP, NDCG and R-precision for both data sets. For TREC Ad Hoc 6-8 it also showed the best results according to the number of relevant retrieved documents.

Tableau 3. *T-test. Significant results are marked by **

		InL2	Bo1	Bo2	CS	KL
TREC 6-8	Mean	0.2105	0.2391	0.249	0.2311	0.24
	Std.Dv.	0.1961	0.2092	0.2152	0.2143	0.21
	Diff.	0.0431	0.0145	0.0045	0.0224	0.0135
	t	6.0666	3.9325	2.3173	3.8832	3.6129
	p	0.0*	0.0001*	0.0218*	0.0001*	0.0004*
	Cnf.-95%	0.029	0.0072	0.0006	0.011	0.0061
	Cnf.+95%	0.057	0.0217	0.0084	0.0338	0.021
WT10G	Mean	0.1893	0.2094	0.219	0.2067	0.2119
	Std.Dv.	0.1865	0.1973	0.2019	0.2014	0.2007
	Diff.	0.031	0.011	0.0013	0.0136	0.0084
	t	3.7126	2.1668	0.3564	3.2594	1.6283
	p	0.0003*	0.0326*	0.7222	0.0015*	0.1066
	Cnf.-95%	0.0144	0.0009	-0.0063	0.0053	-0.0018
	Cnf.+95%	0.0476	0.021	0.009	0.0219	0.0187

For the Web data by this measure our system was right after the Bo1 model. Our system showed the best results for both data sets for P@5, P@10, and P@100. As it was showed in (Savenkov *et al.*, 2011), users trust search engine ranking and they click more frequently on top-ranked documents. Thus, P@5 and P@10 seems to be quite crucial. The differences between the our approach and other evaluated methods are significant at the level $p < 0.05$ for TREC Ad Hoc 6-8 collection. On Web data at $p < 0.05$ the differences with Bo2 and KL models is insignificant. The insignificant difference between LC and Bo2 could be explained by the fact that our method is partially based on Bo2.

7. Bibliographie

Amati G., *Probability Models for Information Retrieval Based on Divergence from Randomness : PhD Thesis*, University of Glasgow, 2003.

Amati G., Carpineto C., Romano G., « Query Difficulty, Robustness, and Selective Application of Query Expansion », *Advances in Information Retrieval*p. 127–137, 2004.

Amini M. R., Tombros A., Usunier N., Lalmas M., « Learning–based summarisation of XML documents », *Inf. Retr.*, vol. 10, n° 3, p. 233-255, 2007.

Buckley C., « Automatic Query Expansion Using SMART : TREC 3 », *In Proceedings of The third Text REtrieval Conference (TREC–3). NIST Special Publication 500–226.*, National Institute of Standards and Technology (NIST), Gaithersburg, MD, p. 69–80, 1995.

Cao G., Nie J.-Y., Gao J., Robertson S., « Selecting good expansion terms for pseudo–relevance feedback », *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, ACM, New York, NY, USA, p. 243–250, 2008.

Carpineto C., Romano G., « A Survey of Automatic Query Expansion in Information Retrieval », *ACM Computing Surveys*, vol. 44, n° 1, p. 1–50, January, 2012.

Cronen-Townsend S., Croft W. B., « Quantifying query ambiguity », p. 104–109, March, 2002.

Ermakova L., « A Method for Short Message Contextualization : Experiments at CLEF/INEX », *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, p. 352-363, 2015.

Gabrilovich E., Broder A., Fontoura M., Joshi A., Josifovski V., Riedel L., Zhang T., « Classifying search queries using the Web as a source of knowledge », *ACM Trans. Web*, vol. 3, n° 2, p. 5 :1–5 :28, April, 2009.

Gabrilovich E., Markovitch S., « Computing Semantic Relatedness using Wikipedia–based Explicit Semantic Analysis », *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*p. 1606-1611, 2007.

Landauer T. K., Foltz P. W., Laham D., « Introduction to Latent Semantic Analysis », *Discourse Processes*p. 259-284, 1998. 25.

Manning C. D., Raghavan P., Schütze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Miao J., Huang J. X., Ye Z., « Proximity-based Rocchio's model for pseudo relevance », *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 535-544, 2012.

Rocchio J., « Relevance Feedback in Information Retrieval », *The SMART Retrieval System*, p. 313–323, 1971.

Savenkov D., Braslavski P., Lebedev M., « Search snippet evaluation at yandex : lessons learned and future directions », *Proceedings of the Second international conference on Multilingual and multimodal information access evaluation*, CLEF'11, Springer–Verlag, Berlin, Heidelberg, p. 14–25, 2011.

Schütze H., « Automatic Word Sense Discrimination », *Computational Linguistics*, vol. 24, nº 1, p. 97-123, 1998.

Singh J., Sharan A., « Co-occurrence and Semantic Similarity Based Hybrid Approach for Improving Automatic Query Expansion in Information Retrieval », *Distributed Computing and Internet Technology. 11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings*, vol. 8956 of *Lecture Notes in Computer Science*, Springer International Publishing, p. 415-418, 2015.

Voorhees E., Harman D., « Overview of the Seventh Text REtrieval Conference (TREC–7) », *Text REtrieval Conference (TREC) TREC–7 Proceedings*, Department of Commerce, National Institute of Standards and Technology, p. 1–23, 1998. NIST Special Publication 500–242 : The Seventh Text REtrieval Conference (TREC 7).

Voorhees E. M., « Query expansion using lexical–semantic relations », *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, Springer–Verlag New York, Inc., New York, NY, USA, p. 61–69, 1994.

Wan J., Wang W., Yi J., Chu C., Song K., « Query Expansion Approach Based on Ontology and Local Context Analysis », *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, nº 16, p. 2839-2843, 2012.

Xu J., Croft W. B., « Query expansion using local and global document analysis », *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, ACM, New York, NY, USA, p. 4–11, 1996.

Xu J., Croft W. B., « Improving the effectiveness of information retrieval with local context analysis », *ACM Trans. Inf. Syst.*, vol. 18, nº 1, p. 79-112, January, 2000.

Xu Y., Jones G. J., Wang B., « Query dependent pseudo–relevance feedback based on wikipedia », *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, ACM, New York, NY, USA, p. 59–66, 2009.

Yang Z., Cai K., Tang J., Zhang L., Su Z., Li J., « Social context summarization », *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, ACM, Beijing, China, p. 255-264, 2011.