# Query Recommendation based terms and relevant documents using language Models

BTIHAL EL GHALI[1], ABDERRAHIM EL QADI[2], OMAR EL MIDAOUI[1], MOHAMED OUADOU[1]
[1] LRIT Associated Unit to the CNRST - URAC n°29, Faculty of Science
Mohammed V-Agdal University Rabat, Morocco
btihal.elghali@gmail.com, omarelmidaoui@gmail.com, ouadou55@gmail.com
[2] TIM, High School of Technology, Moulay Ismaïl University
Meknes, Morocco
elqadi_a@yahoo.com

*Abstract:* - The query submitted by the user is only a partial and often ambiguous expression of his need. This implies that it is essential to suggest to the users the most related queries to the context of their queries. However, the notion of context is quite broad and includes all the elements related to the query (Its field, its environment, the user profile, his preferences and his search history). In this paper, we extract the environment of a user's query in order to use it later in its query recommendation process. For this purpose, three different methods of query recommendation are proposed, and then compared based on the quality of the extracted environments, by calculating the Average Internal Similarity (AIS) of each built environment. The results show that the information of documents relevance influence the similarity between queries better than the information of existence of terms for all the proposed approaches. The final experiment was a comparison between the three approaches, and it shows that for short and long queries the highest value of AIS is reached by the TLM approach using Language Models based on common terms and relevant documents.

*Key-Words:* - Information Retrieval, Query Recommendation, Language Model, Recommendation Algorithm, Query's context.

## 1 Introduction

Query recommendation is the fact of suggesting queries that are almost similar to the user query, and it can be considered as a method for improving retrieval performance.

As queries get longer, there are more possibilities that some important terms co-occur in the query and the relevant documents [1]. However, a study was made on the user's queries on a search engine [2], and it was observed that users usually submit very short queries. The average length of web queries is two words. Moreover, these queries most of the time contain ambiguous terms. Thus, using the initial user query to retrieve relevant documents is an almost impossible task [3], because of the increasing volume of information bases.

Indeed, the query is only a partial and often ambiguous expression of the user's information needs. Considered separately, it is insufficient to clearly identify what the user is looking for. Considering the context, the partial information of the query can be completed and the ambiguity can be resolved to a certain degree [4].

For a correct interpretation of the user's query, it has been demonstrated that it should be placed in its appropriate context [5]. The context is a large notion that includes the user context (his domains of interest, his preferences and his historic of research) and the query context, which mean the environment of the query (its relevant documents and its terms). The first context needs the research to be done using users profiles, but a single profile can group a large variety of domains and interests, that are not always relevant for a particular query [5]. Thus, the solution is to use the second context as an appropriate context to improve the precision of the query. The creation of multiple profiles is also a possible solution to this problem according to [6], one profile for each domain of interest. Then for every new query, a single domain is identified.

To cover the gap between the original user query and his need of information many methods was proposed. The most common methods used are query reformulation, query expansion, and query recommendation techniques. Traditionally, the concept of language model [4][7][8][9][10][11] are

exploited in the field of information retrieval in order to represent the relation of relevance between a document and a query, by estimating the probability of generation of the query by the language model of the document.

The main objective of this work is to provide high-level suggestions for the original user query. We consider the query context that we build using queries extracted from a log of past queries.

We propose to use the language model for estimating the probability of generation of the user query by the language model of the past user queries archived in the query logs of the search engine. Thus, we compute the score of recommendation of the past queries to the new user query, by using the score function defined in Kullback Leibler divergence (KL-divergence) [12] for language models, based on queries terms and queries shared relevant documents.

We estimate that the quality of recommendation using language models is better than the quality of recommendation using the Query Recommendation Algorithm proposed in [13], even after improving it by applying to it some modifications, concerning the weighting function and the similarity measure expressions used [14]. Thus, in this paper we varied all the parameters used in the different formulas used in these techniques, in order to found the value of each parameter that gives the best result. Then we compare these methods by varying the technique of score computation based on queries terms or on queries relevant documents.

Experiments have been performed on the database CISI of the collection SMART. Intensive experiments have been done to select the suitable parameters to use on the score of recommendation and for the language model Smoothing [9].

This paper is structured as follows: The section 2 presents the related works. In section 3, we describe the Query Recommendation Algorithm proposed. Section 4 gives a brief definition of language model and their smoothing and describes the method proposed based of them. Section 5 shows our experimental results. Finally, section 6 summarizes the main conclusions of this work and gives a brief idea on our future works.

## 2 Related Work

Search engines' logs keeps track of queries and URLs selected by the users when they are finding useful data through the search engines.

In order to enhance keyword-based queries, query recommendation is considered an effective assistant in search engines to suggest related queries for users when the results of an initial input query are not sufficient. In addition to that, the past queries stored in query logs can be a source of additional evidence to help future users in improving search quality. By combining these two notions was created a process to help people to refine their search queries based on other people's similar queries, by exploiting the information about users' interactions contained in the search engine log.

Ricardo Baeza-Yates [15] proposes a method for suggesting a list of queries that are related to the user input query, based on clustering similar past queries. He hypotheses that semantically similar queries may not share query-terms but they do share terms in the documents selected by users. Thus, the query recommendation algorithm presented in this work operated in three steps. First, past users' queries are represented in a term-weight vector with the text of their clicked URL's, and separated into clusters. The second step is processed when a new query is submitted to the search engine, by finding the cluster to which it belongs and then compute a rank score for each query in the cluster according to the new query. Finally, the related queries are returned ordered according to their rank score.

Hamada M. Zahera [13] presents the same algorithm's steps by modifying the definition of the query vectors used in the clustering step and proposing a new similarity measure called Tanimoto Coefficient to rank the related queries. The experiments of these two algorithms over a real query logs shows their effectiveness.

Rinki Khanna and Asha Mishra [24] had also proposed a query recommendation scheme towards better information retrieval to improve search engine effectiveness to a large scale. The algorithm presented in their work, exploit the information about user activities through the search results, which are extracted from query logs. Then, a clustering process is applied using the similarities based on Keyword and Clicked URL's. In order to discover the most favored queries within every query cluster and suggest them as recommended queries for the user's new query.

It is clear that related queries can be found by examining the collection of queries and URLs in a bipartite graph, whose edges are connecting past users' queries to the corresponding URLs returned

by a search engine. Lin Li has described a method called QUBIC in [16], which considers queries that are more strongly connected to each other in a query-URL bipartite graph as more similar. The QUBIC system, construct a query-URL bipartite graph using a query-URL historical collection and also construct a query affinity graph (QAG) using query-URL based similarity measures in order to reduce the set of candidate queries. Two queries are defined as related if there are paths from one to another. Then, connected queries in the QAG are extracted as a group of queries with high similarity. A system-defined parameter $\delta$ has been introduced to control the level of query similarity to be considered. The last step is the ranking of similar queries by taking into account the propagation of the similarity in the QAG. Finally, the experimental results using "Query KDD Data Set" and "Query TREC Data Set" demonstrate the effectiveness and feasibility of the QUBIC approach.

Recently, researchers in the information retrieval field integrate the semantic aspect to query expansion, document ranking and clustering, question-answer systems and query recommendation. In Lingling Meng's research, a new model for similarity metric of web queries was presented [17] using a query log. The model takes into account both, word form of the two queries and their semantic information. As a thesaurus that focuses on word meaning instead of word forms, WordNet is used to obtain the semantic information. The presented approach uses the bottom-up hierarchical clustering method, in order to cluster past users queries to find the different topics contained in the search engine's log. The clustering process is applied based on the new similarity metric proposed. The new model shows it's good performance in improving the query expansion recall of 8,1% and precision of 9,2%.

User's queries archived on query logs, can also be used to construct an Aggregate Markov Chain (AMC) through which the relevance between the keywords seen by the system is defined. This AMC is used in the Markovian Semantic Indexing (MSI) [22], which is a method for annotation based image retrieval that is particularly suitable for the Annotation-Based Image Retrieval (ABIR) tasks when the image's annotation data are limited. Then, the images are ranked based on markovian distance, based on the probability of matching between annotated images and user queries. Finally, the images which has the maximum probability to match with the new user query are retrieved.

This Markov Chain based method is efficient as compared to all previous methods of image retrieval. However, the resulted images do not satisty the end users for sure. This is one of the limitations of this method. Therefore, the work presented by [23] is extending this method with goal of achieving the end user satisfaction and improve further precision and recall rates.

# 3 Query Recommendation Algorithm

Our first contribution in this work is the modification of the Query Recommendation Algorithm (QRA) presented in [13]. In a previous work [14], we eliminated the steps 1 and 2 of the algorithm that concern the clustering of the past queries and the identification of the appropriate cluster of a new query when submitted. We also changed the weighting function and the similarity measure based on many comparisons. Compared to [14], in this paper, we also eliminated the third step of the QRA, because, we believe that the support of a query do not give any information about the degree of relatedness between two queries.

In order to suggest related past queries to an input query, we represent each query with a term-weight vector and a document-weight vector. Then, we measure similarities between the input query and each past query using term vectors and document vectors. In the next step, the interest of a related query is measured by its rank to the input query, by normalizing and combining the term-based similarity with the document-based similarity. Finally, the past queries extracted from the search engine's log are classified according to the new user query.

The objective of the measuring of similarities between the queries, represented as a document vectors, is to search for queries that have many common relevant documents. In the other hand, the aim of computing similarities between queries, using their term vectors representation, is to search for queries that have an important number of common terms, to consider that these queries are associated depending on the terms that they contain.

The details of the Query Recommendation Algorithm steps are as follows:

1-We build two vectors for each query $Q_j=\{D_1^{(j)}, D_2^{(j)}, \ldots, D_l^{(j)}\}$ and $Q_j=\{T_1^{(j)}, T_2^{(j)}, \ldots, T_m^{(j)}\}$, where $D_i^{(j)}$is the weight of the $i^{th}$ document in the documents vector, and $T_i^{(j)}$ represents the weight of the $i^{th}$ term for the same query in the terms vector.

These weights are defined by classic weighting measure LTC [18]:

$$D_i^{(j)} = \frac{\log(tf_i{}^{(j)}+1)\times idf_j}{\sqrt{\sum_{k=1}^{n}[\log(tf_k{}^{(j)}+1)\times idf_j]}} \qquad (1)$$

with: $idf_j = log(N/n_j)$

With $tf_i^{(j)}$ is the frequency of the j[th] document in the set of relevant document for the query $Q_j$, N is the total number of queries in the log and $n_j$ is the number of queries for which the j[th] document is relevant.

We calculate each $T_i^{(j)}$ with the same expression.

2- We measure the similarities between the new query of the user $Q_n$ and each past user query $Q_p$ based on the two representations by using the expression of similarity Cosine.

The Cosine measure calculates the similarity between two vectors by determining the angle between them, and its expression is:

$$Sim(Q_n, Q_p) = \cos(\vec{q}_n, \vec{q}_p) = \frac{\vec{q}_n \times \vec{q}_p}{|\vec{q}_n| \times |\vec{q}_p|} \qquad (2)$$

This expression consider that two objects (queries for example) are similar if their vectors are confounded [19].Otherwise, the two objects are not similar and their vectors form an angle $(\vec{q}_n, \vec{q}_p)$, whose cosine is the value of the similarity.

3- We compute the rank of each past query for the input query using the two representations of each query. The ranking score of the query $Q_p$ for the initial query $Q_n$ is measured as follow:

$$Rank(Q_n, Q_p) = \gamma Sim_T(Q_n, Q_p) + (1-\gamma) Sim_D(Q_n, Q_p) \quad (3)$$

Where $Sim_D(Q_n, Q_p)$ is the similarity between the queries $Q_n$ and $Q_p$ based on the documents representations, $Sim_T(Q_n, Q_p)$ is the similarity between the same two queries based on their term-weight vectors and $\gamma \in [0,1]$ is a parameter that we used for normalization.

Our contribution in this algorithm takes into account also the fact that we use the similarity based on terms in the expression of the rank.

4- We classify the past queries according to the input query based on their ranking score.

# 4 Query Recommendation using Language Models

In information retrieval, the basic principle of language models is to order the documents of a collection according to their ability to generate the user query. Thus, the relevance of a document to a query is bound to the fact that the language model (LM) of the document can generate the language model of the query.

In this paper, we propose to represent the relation of recommendation between two queries using language models. We order past queries $Q_p$ according to their capacity to generate the new user query $Q_n$.

The ranking function that we used is the typical score function defined by *KL*-divergence in the language modeling framework. Whose expression is as follows [5][20]:

$$Score_{LM}(Q_n, Q_p) = \sum_{t \epsilon V} P(t|\theta_{Q_n}) \log(P(t|\theta_{Q_p}) \approx -KL(\theta_{Q_n} || \theta_{Q_p}) \qquad (4)$$

Where $\theta_{Q_n}$ is the language model of the new query, $\theta_{Q_p}$ the language model created for a past query, and V the vocabulary of terms.

$P(t|\theta_Q)$ represents the probability of a term t in the language model of the query and are measured using the Maximum Likehood Estimation (MLE), as follows:

$$P(t|\theta_Q) = \frac{f(t)}{\sum_{t_i \epsilon Q} f(t_i)} \qquad (5)$$

With $f(t)$ is the frequency of t in the query.

The main problem that occurs for language models is the under-representation of data. Because the size of the training corpus cannot reach the size of a language. Thus, the absent terms in the training corpus are estimated by a null probability. Therefore, a null probability is assigned to any sequence of words containing that word.

The proposed solution to this problem is the "Smoothing". Which aim is to assign a not null probability to the absent terms from the training corpus, by redistributing the probability mass observed. Several smoothing methods have been developed [9].The choice of the appropriate smoothing technique depends on the environment of experimentation according to [10]. One of the common smoothing methods used in information retrieval is the Jelinek-Mercer interpolation smoothing:

$$P\left(t\middle|\theta'_{Q_p}\right) = (1-\lambda) P\left(t\middle|\theta_{Q_p}\right) + \lambda P(t|\theta_C) \quad (6)$$

Where $\lambda$ is an interpolation parameter and $\theta_C$ the language model of the collection of queries extracted from the search engine log.

# 5 Experimental results

In order to verify the performance of the approaches proposed above, we compared three proposition of fusion between them, by varying the score's part

based on terms or the part based on documents pertinence.

The propositions to compare are:

- TQRA: The technique of recommendation based totally on the Query Recommendation Algorithm (QRA) presented in section 3, using the equation 3.

- LM-QRA: The method of recommendation using Language Models (LM) based on terms and using the QRA based on documents:

$$Rank(q_j, q_i) = \gamma Score_{LM-T}(q_j, q_i) + (1-\gamma) Sim_D(q_j, q_i) \quad (7)$$

- TLM: The technique based totally on the score of recommendation using LM for terms and documents:

$$Rank(q_j, q_i) = \gamma Score_{LM-T}(q_j, q_i) + (1-\gamma) Score_{LM-D}(q_j, q_i) \quad (8)$$

The measure that we used to evaluate and to compare these propositions is the "Average Internal Similarity" (AIS). This measure consider a cluster of vectors and calculate the similarity between them. In our work, we consider each input query and its recommended queries as a cluster. The AIS is computed as follows [21]:

$$AIS(c) = \frac{sum\,(c)^2 - |c|}{|c|(|c|-1)} \quad (9)$$

With c the cluster of queries, |c| the number of vectors (queries) in the cluster and sum(c) is a vector, which represents the sum of all the vectors in a cluster c.

In each experiment, we used terms vectors or/and documents vectors to measure the AIS value of a group of queries. Thus, we define three different AIS: $AIS_T$ using terms vectors, $AIS_D$ using documents vectors and $AIS_A$, which is the average of $AIS_T$ and $AIS_D$.

As a collection of test, we used the database CISI from the standard collection SMART, which is a database of queries and documents in Library Science and related areas. This collection provides 111 queries, 1460 documents and a matrix representing the relevance or non-relevance of each document to each query.

Using short queries (contain less than 5 terms) and long queries (contain more than 5 terms). We tested our approaches in the queries of the database CISI.

To realize our experiments, we developed a java applications for the pretreatment of the queries' text (Stop words elimination, Stemming), to do the Query Recommendation Algorithm and Language Model calculations, and to compute the "Average Internal Similarity" of each group of queries.

Before comparing the three approaches presented above, we have to identify the best value to use for some parameters. First, we varied the parameter $\gamma$ in the equation 3 from 0 to 1 to identify the importance of the score based on terms and the score based on documents.

The $AIS_A$ is computed for clusters of 6 queries, each input query with its 5 best recommended queries, using documents and terms vectors.

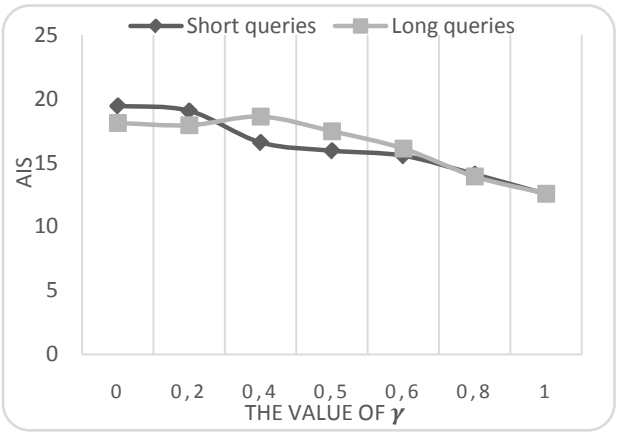


Fig.1. Variation of the parameter $\boldsymbol{\gamma}$ using the TQRA.

We notice in figure 1, that for short queries the best value of the $AIS_A$ (19,44) is given using $\boldsymbol{\gamma}$ = 0, we can explain that easily by the fact that short queries contain less than 5 terms, and the similarity using the statistical measure cosine is not important using terms vectors. Thus, the best group of queries is constructed with the input query and its five best recommended queries using only the score based on documents vectors.

When increasing the influence of documents relevance on the score of query recommendation (going from $\boldsymbol{\gamma}$=1 to $\boldsymbol{\gamma}$=0) the performance of the recommendation is also increasing, while generally the existence of terms influences on it negatively. Despite this, we believe that using methods based on terms are indispensable in the domain of information retrieval. Thus, we decided to consider the $\boldsymbol{\gamma}$ = 0,2 for short queries in what follows because of its nearness from the first value (19,05).

Regarding the input queries that have an important number of terms, the effectiveness of the approach is better using term vectors. We notice that for long queries the best value of $AIS_A$ is 18,6 and it is given using $\boldsymbol{\gamma}$ = 0,4.

The second step of experimentations is done in order to identify the best value of $\lambda$ for the smoothing (equation 6) in Language Models based

on terms. In this step also, we used the 5 best recommended queries only to calculate the $AIS_T$ and we varied $\lambda$ from 0 to 0,8 because by using the value 1 the equation 6 become: $P\left(t\middle|\theta'_{Q_p}\right) = P(t|\theta_C)$. Thus, for a term t the value of $P\left(t\middle|\theta'_{Q_p}\right)$ is the same for all the past queries. In fact, the score of recommendation to a new query is also the same for all the past queries.

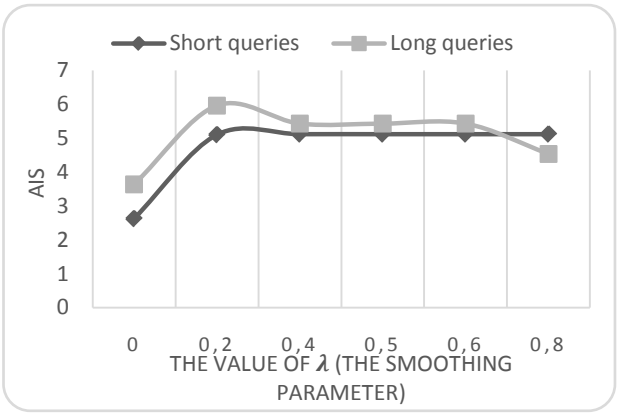


Fig.2. Variation of the smoothing parameter $\lambda$ in the language model score based on terms.

Figure 2 shows that for short queries the $AIS_T$ increases from a low value for $\lambda$=0 to its best value when smoothing with 0,2 and keeps it until $\lambda$=0,8. While, for long queries the best value of $AIS_T$ is reached at 0,2 and then increases until having a lower value at 0,8. Thus, we can conclude that using language models based on terms for query recommendation with the parameter of smoothing equal to 0,2 is enough to reach the best values of $AIS_T$ using our collection of test.

Then using the 5 best recommended queries for each input query in every case, and considering $\boldsymbol{\gamma}$ = 0,2 for short queries and $\boldsymbol{\gamma}$ = 0,4 for long queries, we proceed to the comparison of the three approaches: TQRA, LM-QRA, and TLM. The comparison is represented in table 1, figure 3(a) and figure 3(b).

In figure 3 (a and b), we chose four short queries and four long queries randomly to compare the three proposed approaches more precisely. While, in Table 1 we present the average of $AIS_A$ for all the short queries and long queries.

| $AIS_A$ | TQRA | LM-QRA | TLM |
|---|---|---|---|
| **Short queries** | 19,05 | 16,64 | 19,87 |
| **Long queries** | 18,60 | 17,30 | 19,04 |

Table 1. $AIS_A$ for short and long queries using the three proposed approaches

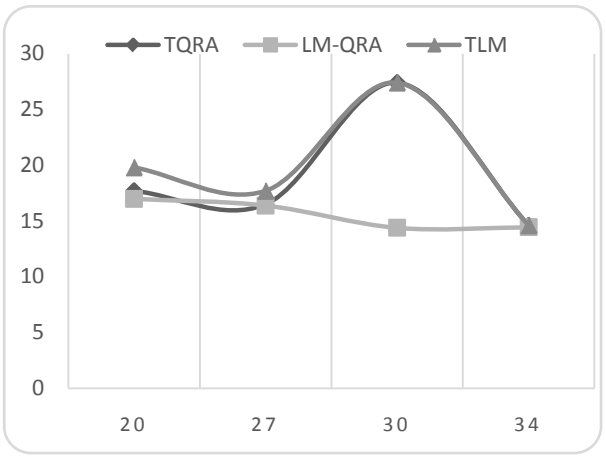


Fig.3(a). Comparison of the $AIS_A$ of some short queries for the three approaches using $\boldsymbol{\gamma}$=0,2

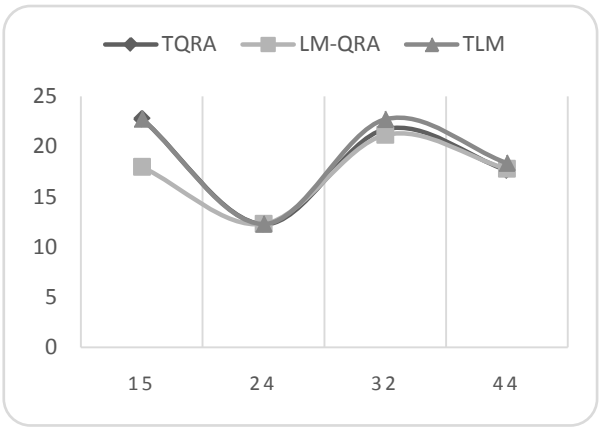


Fig.3(b). Comparison of the $AIS_A$ of some long queries for the three approaches using $\boldsymbol{\gamma}$=0,4

The figures 3(a), 3(b) and table 1, show that for short and long queries the highest value of $AIS_A$ is reached by the third approach (TLM) which uses Language models for the two representations of the queries. We notice also that the value of $AIS_A$ using the TQRA approach is quite near to the $AIS_A$ of the TLM approach, and is equal to it in some cases (queries number 30, 34 and 15). While, the LM-QRA approach that is the fusion of the language Model approach using terms vectors and the Query Recommendation Algorithm using documents vectors, gives a low value in the two cases (short and long queries).

With these results, once again the language models show their performance in Information Retrieval, while they are used for the first time in Query Recommendation.

In the last step of experiments, we proceed to the comparison of the three approaches by varying the number of recommended queries from 1 recommended query to 10 recommended queries.

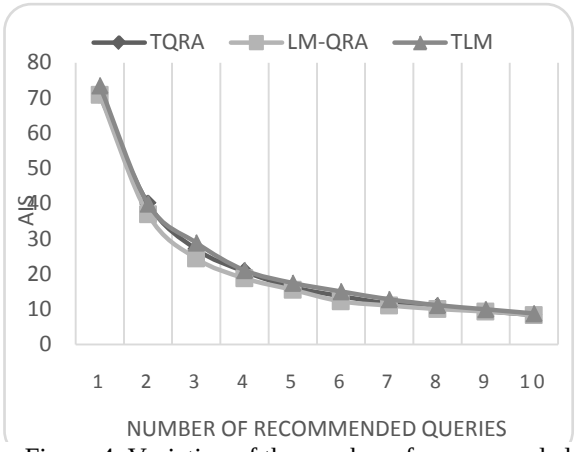

Figure 4. Variation of the number of recommended queries for the three approaches

In figure 4, we notice that the $AIS_A$ is lower when using more recommended queries. That is because the number of vector in the cluster biases the expression of AIS. We notice also, that while varying the number of recommended queries, the TLM approach is still the approach, which gives the highest results.

# 6 Conclusion

In this work, we have proposed three different approaches of query recommendation in order to compare them to improve the process of suggestion of queries to the search engines users. Our propositions are based on different methods and uses the information of documents relevance to queries and terms existence in queries (the new user's query and the past users' queries). We integrated the notion of Language Models for the first time to the query recommendation process. We did our experimentations using short queries and long queries of the database CISI from the standard collection of test SMART. The results show that the information of documents relevance influence the similarity between queries better than the information of existence of terms. Our experiments show also that for short and long queries, the TLM approach, which uses the Language Models notion based on terms and documents, gives the highest values of AIS, with an improvement of 4,3% for short queries, and 2,37% for long queries compared to the TQRA approach.

*References:*

[1] Xu, J., Croft, W.B., Improving the effectiveness of information retrieval with local context analysis, *ACM Transactions on Information Systems (TOIS)*, Vol. 18, No. 1, January 2000, pp. 79-11.

[2] Wen, J., Nie, J., and Zhang, H., Clustering User Queries of a Search Engine, *In Proceedings of WWW10*, Hong Kong, May 2001.

[3] Baziz, M., Indexation conceptuelle guide par ontologie pour la recherche d'information, *PhD thesis, Institut de Recherche en Informatique de Toulouse*, Paul Sabatier University, Toulouse, France, December 2005.

[4] Bouchard, H., Nie, J.-Y., Modèles de langue appliqués à la recherche d'information contextuelle. *In CORIA'06*, 2006, pp. 213-224.

[5] Bai, J., Nie, J-Y. Bouchard, H., Cao, G., Using query contexts in information retrieval, *SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM New York, USA, 2007, pp. 15-22.

[6] Liu, F., Yu, C., Meng, W., Personalized web search by mapping user queries to categories, *In CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, November 2002, pp. 558-565.

[7] Ganguly, D., Leveling, J., Jones, G. J.F., Query expansion for language modeling using sentence similarities, *In: The 2nd Information Retrieval Facility (IRF) Conference*, Vienna, Austria, June 2011.

[8] Bai, J., Nie, J-Y., Using Language Models for Text Classification, *ACM Conference'04*, Washington D.C., U.S.A, Month 11, 2004.

[9] Chen, S.F., Goodman, J., An Empirical Study of Smoothing Techniques for Language Modeling, *Technical Report TR-10-98, Computer Science Group*, Harvard University, Cambridge, Massachusetts, 1998, pp. 310-318.

[10] Cao, G., Nie, J., Bai, J., Integrating Word Relationships into Language Models, *In Proceedings of SIGIR'05*, Salvador, Brazil, August 2005.

[11] Zhai, C., Statistical Language Models for Information Retrieval: A Critical Review, *Foundations and Trends in Information Retrieval*, Vol. 2, No. 3, 2008, pp. 137-215.

[12] Imran, Ha.,Sharan, A., Selecting Effective Expansion Terms for Better Information Retrieval, *International Journal of Computer Science & Applications (IJCSA)*, Vol. 7, No. 2, 2010, pp. 52-64.

[13] Zahera, H.M., El Hady, G.F., Abd El-Wahed, W.F., Query Recommendation for Improving Search Engine Results, *International Journal of Information Retrieval Research*, Vol. 1, Issue 1, January 2011, pp. 45-52.

[14] El Ghali, B., El Qadi, A., El Midaoui, O.,Ouadou, M., Aboutajdine D., Probabilistic Query Expansion Method based on a Query Recommendation Algorithm, *International Journal of Web Applications (IJWA)*, Volume 5, Number 1, March 2013, pp. 1-12.

[15] Baeza-Yates, R., Hurtado, C., Mendoza, M., Query Recommendation Using Query Logs in Search Engines, *Current Trends in Database Technology - EDBT 2004 Workshops*, Lecture Notes in Computer Science, Volume 3268, 2005, pp. 588-596.

[16] Lin, Li., Yang, Z., Liu, L., Kitsuregawa, M., Query-URL Bipartite Based Approach to Personalized Query Recommendation, *Association for Advancement of Artificial Intelligence AAAI'08 Proceedings of the 23rd national conference on Artificial intelligence*, Vol. 2, 2008, pp. 1189-1194.

[17] Meng, L., Huang, R.,Gu, J., A New Model for Measuring Similarity of Web Queries and Its Application in Query Expansion, *International Journal of Grid and Distributed Computing*, Vol. 6, No. 4, August 2013, pp. 51-62.

[18] Kjersti, A., Eikvil, L., Text Categorisation: A survey, *Technical Report, Norwegian Computing Center*, Norway, June 1999.

[19] Slimani, T., Ben Yaghlane, B., Mellouli, K., Une extension de mesure de similarité entre les concepts d'une ontologie, *In Proceedings of SETIT 2007*: *4rth International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, TUNISIA, 25-29 March 2007.

[20] Asfari, O.,Doan, B-L., Bourda, Y., Sansonnet, J-P., Context-based Hybrid Method for User Query Expansion, *In Proceedings of the fourth International conference on Advances in Semantic Processing (SEMAPRO)*, Italy, Florence, 2010, pp. 69-74.

[21] O'Connor, B., Clustering Political Words: Senses and Connotations. *CS224N / Ling 237*. Final Project, June 5, 2003.

[22] Raftopoulos, K., Ntalianis, K., Sourlas, D., S. Kollias, S., Mining User Queries with Markov Chains: Application to Online Image Retrieval, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 2, February 2013, pp. 433-447.

[23] Walunj, V.S., Patil, S.R., Online Image Retrieval Based on Relevance Feedback and Markov Chain for Mining User Queries, *International Journal of Emerging Technology and Advanced Engineering*, Volume 4, Issue 6, June 2014, pp. 558- 562.

[24] Khanna, R., Mishra, A., A Survey on Advanced Page Ranking in Query Recommendation, *International Journal of Computer Science and Mobile Computing*, Vol. 3, Issue. 4, April 2014, pp. 989 – 995.