ORIGINAL RESEARCH

A query suggestion method combining TF-IDF and Jaccard Coefficient for interactive web search

Suthira Plansangket [*], John Q Gan

School of Computer Science and Electronic Engineering, University of Essex, United Kingdom

Abstract

This paper proposes a query suggestion method combining two ranked retrieval methods: TF-IDF and Jaccard coefficient. Four performance criteria plus user evaluation have been adopted to evaluate this combined method in terms of ranking and relevance from different perspectives. Two experiments have been conducted using carefully designed eighty test queries which are related to eight topics. One experiment aims to evaluate the quality of the query suggestions generated by the proposed method, and the other aims to evaluate the improvement of the relevance of retuned documents in interactive web search by using the query suggestions so as to evaluate the effectiveness of the developed method. The experimental results show that the method developed in this paper is the best method for query suggestion among the methods evaluated, significantly outperforming the most popularly used TF-IDF method. In addition, the query suggestions generated by the proposed method significantly improve the relevance of returned documents in interactive web search in terms of increasing the precision or the number of highly relevant documents.

Key Words: Query suggestion, Query expansion, Information retrieval, Search engine, Performance evaluation

1 Introduction

Internet search engines play the most important role in finding information from the web. One of the great challenges faced by search engines is to understand precisely users' information need, since users usually submit very short (only a couple of words) and imprecise queries.[1] Most existing search engines retrieve information by finding exact keywords. Sometimes, users do not know the precise vocabulary of the topic to be searched and they do not know how search algorithms work so as to produce proper queries.[2]

One solution to these problems is to devise a query suggestion module in search engines, which helps users in their searching activities. Kelly *et al.*[3] pointed out that query suggestions were useful when users ran out of ideas or faced

a cold-start problem. Kato *et al.*[4] analysed three types of logs in the Microsoft's search engine Bing and found that query suggestions are often used when the original query is a rare query or a single-term query or after the user has clicked on several URLs in the first search result page. Furthermore, Carpineto and Romano[5] found that one advantage of query expansion is that there is more chance for a relevant document that does not contain the original query terms to be retrieved. Niu and Kelly[6] found that when searching for the most difficult topic, users save significantly more documents retrieved by query suggestions than by user-generated queries. There exist many query suggestion methods that extract query related terms or features from log files, ontologies, and documents returned from search engines and use them to generate query suggestions.

More related work will be described in the next section.

This paper proposes a query suggestion method combining two ranked retrieval methods: TF-IDF and Jaccard coefficient, and evaluates the method using several performance criteria and users' judgement as well in terms of the quality of the generated query suggestions and the improvement of relevance of the returned documents in interactive web search. Comprehensive comparative experiments have demonstrated the effectiveness of the method developed in this paper.

2 Related work

2.1 Query expansion and reformulation

Query expansion is a technique to expand the query with related words and is widely used for query suggestion. It aims to improve the overall recall of the relevant documents.[7, 8] Query reformulation or dynamic query suggestion is more complex than query expansion, which forms new queries using certain models.[8–10] This paper mainly addresses query expansion.

2.2 Explicit and implicit feedback

Relevance feedback plays an important role in query suggestion. There are two major categories of relevance feedback. Explicit feedback is provided directly by users, which is expensive and time consuming. On the other hand, implicit feedback is derived by the system.[7] The system derives the feedback information from several sources of features, such as log files, web documents, and ontologies. This paper focuses on query suggestion methods based on implicit relevance feedback.

There are many studies on query suggestion using log files, from which user's search behaviours and information need can be derived.[1, 4, 8, 11–18] Various ontologies have been applied to create knowledge-driven models for generating query suggestions, such as WordNet,[19, 20] Wikipedia,[21] ODP and YAGO.[22–25] Query suggestions can also be generated from query related features extracted from web documents returned by search engines.[2] There are some studies on query suggestion that combined query log and web search results[26] or combined query log and ontology.[27]

2.3 Ranked retrieval models

In ranked retrieval models, the system returns an ordered list of top matching documents with respect to a query. Typical ranked retrieval methods include Jaccard coefficient and Term frequency - inverse document frequency (TF-IDF)[28] which will be described in more detail in the next section.

In information retrieval, ranked retrieval methods are used to order relevant documents with respect to a query. Similarly, highly relevant query suggestions are preferable to

appear first in query suggestions.[29] Therefore, it is reasonable to adapt ranked retrieval methods for query suggestion.

3 Methods

3.1 TF-IDF

TF-IDF[7] is the most popular term weighting scheme in information retrieval. The TF-IDF score of a term in a set of documents is calculated as follows:

$$tfidf_i = \sum_{j=1}^{N} w_{ij} \qquad (1)$$

$$w_{ij} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i}, & if \ f_{i,j} > 0 \\ 0, & otherwise \end{cases} \qquad (2)$$

where $f_{i,j}$ is the frequency of term i in document j, n_i is the number of documents in which term i appears, N is the total number of available documents.

TF-IDF has been used to measure word relatedness.[30] Therefore, it can be applied to identify terms in the documents returned from search engines, which are mostly relevant to the original query, as query suggestions.

3.2 Jaccard coefficient

Jaccard coefficient[28] is a measure of overlap of two returned documents D1 and D2, which are represented as vectors of terms and may not have the same size. Jaccard coefficient has been used to measure the similarity between search texts.[8] Kulkarni and Caragea[31] used this method to compute semantic relatedness between two concept clouds.

The Jaccard coefficient for a length-normalized model is calculated as follows:

$$Jaccard(D_1, D_2) = \frac{|D_1 \cap D_2|}{|\sqrt{D_1 \cup D_2}|} \qquad (3)$$

where $\cap$ represents intersection and $\cup$ union. In this paper, D_1 and D_2 are bags of words which contain query suggestion candidates that are selected from words which appear in at least two returned documents. In mathematics, the notion of multiset or bag is a generalization of the notion of set, in which members are allowed to appear more than once. The intersection or union of multisets is a multiset in general.[32]

If a query suggestion candidate is from more than two returned documents, its Jaccard coefficient can be extended as follows:

$$Jaccard(D_1, D_2, \cdots D_M) = \frac{|D_1 \cap D_2 \cap \cdots D_M|}{|\sqrt{D_1 \cup D_2 \cup \cdots D_M}|} \qquad (4)$$

In this paper, for each query suggestion candidate, M documents that contain this suggestion term are identified, and then Jaccard coefficient is calculated as the score to rank this candidate.

3.3 A new method based on the combination of TF-IDF and Jaccard coefficient

Our initial experiment found that the TF-IDF method was capable of producing suggestions relevant to the user's original query whilst Jaccard coefficient was good to rank the suggestions. Therefore, a method called Tfjac is proposed in this paper, which selects terms from the combination of the top ten candidate words from the TF-IDF method and up to ten candidate words from the Jaccard coefficient method. The process starts with finding duplicate words from both methods. If the number of these words is less than ten, more candidate words from the Jaccard coefficient method are added. If the number of terms is still less than ten, more candidate words from the TF-IDF method are added till ten query suggestions are selected. The overall process of the Tfjac method is shown in Figure 1.
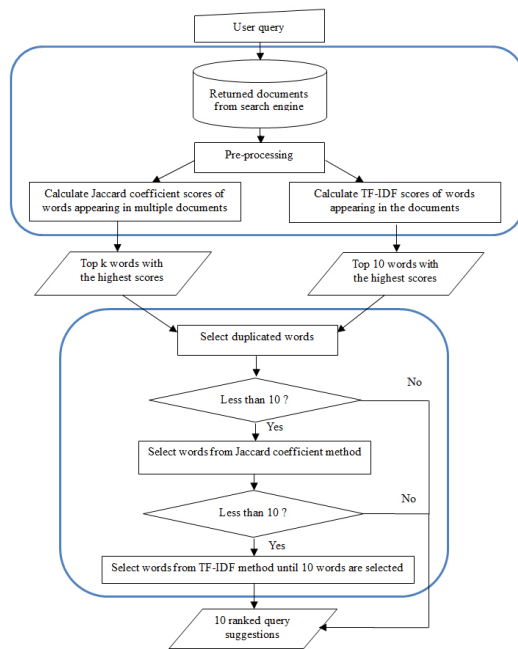

Figure 1: Diagram of the Tfjac method

4 Experiments and results

4.1 Experimental design

He and Ounis[33] proposed an entropy measure which estimates how the occurrences of a query term spread over returned documents. The higher the entropy is, the more a returned document is related to the query. Their results show that the entropy in the top five returned documents is very high, and it decreases rapidly in the remaining documents. Therefore, it has been decided that in this experiment query suggestions are created from analysing the top eight Google search returned documents. That would be enough to generate highly relevant or good suggestions to the original query from these documents. Each document is pre-

processed as follows. First of all, not the whole document, but only the title and snippet content in each document are considered. After that, all HTML tags are removed and all contents are separated into tokens. Thirdly, since the most selective terms for query suggestions should be nouns,[7, 34] only nouns are considered for suggestions.

There are two experiments in this paper. The first experiment aims to evaluate the quality of the query suggestions generated by the method combing TF-IDF and Jaccard coefficient, in comparison with the TF-IDF and Jaccard coefficient methods respectively. A simple search engine using the Google API has been implemented in this experiment. From the analysis of titles and snippets of the top eight Google returned documents using the query suggestion methods, query suggestions are generated by each method for each query. The second experiment aims to evaluate and compare the relevance of the returned documents in interactive web search, which are retrieved from the original query with and without using query suggestion respectively so as to evaluate the effectiveness of the proposed query suggestion method.

Table 1: Categories of test queries

Category	Description	Number of queries
1	Movies	10
2	Food	10
3	Traveling	10
4	Shopping	10
5	Sports	10
6	Arts	10
7	Flowers	10
8	Animals	10
Total		**80**

For evaluation purposes, eighty test queries were selected from eight popular search topics (categories), as shown in Table 1. Each category contains ten queries consisting of one to three words that are commonly known and convenient for user evaluation. These test queries were used in both experiments. It is important to know whether a query suggestion is truly good or not in the performance evaluation. In the first experiment, highly relevant, mildly relevant and irrelevant suggestions for each test query were judged by two approaches in order to reduce subjective bias to expected results and make the experimental results more reliable. Fifty percent of the decisions were based on the suggestions by the Google search engine, which has been widely recognized, and another fifty percent of the decisions were made by users who were five PhD students participating in this experiment. In the second experiment, highly relevant and mildly relevant websites returned from the Google search for each test query were judged by eight participants from the University of Essex. For the performance evaluation, four evaluation methods: Mean reciprocal rank(MRR), Mean Average Precision(MAP), precision at 10(P@10), and

discounted cumulated gain(DCG), were used in both experiments.

4.2 User's selection of suggested queries and assessment of relevance of search results

To confirm whether the evaluation results are acceptable by real users, user evaluation has been conducted as well. Questionnaires were used to obtain users' evaluative feedback. Five participants who were PhD students studying in different fields were asked to select a top suggestion from the query suggestions made by each query suggestion method for each of the eighty test queries and then rank the top suggestions made by the three query suggestion methods for each test query from one to three.

For evaluating the relevance of the search results, questionnaires were also used. In each questionnaire, there are sixteen returned websites for each test query, half of which were returned by using the original query and the other half returned by using query suggestion. The eight participants were asked to select and rank the top three most relevant web sites for each of the eighty test queries from one to three.

4.3 Performance criteria

Four performance criteria have been used in the experiments: MRR, MAP, DCG, and P@10.

4.3.1 Mean reciprocal rank

MRR[35, 36] is a statistic measure suitable for query suggestion's ranking evaluation. For query j, the reciprocal rank of a good query suggestion i, RR_{ji}, is the multiplicative inverse of the rank of this suggestion in the list of potential query suggestions made by a query suggestion method, r_{ji}, i.e.,

$$RR_{ji} = \frac{1}{r_{ji}} \qquad (5)$$

MRR is the average of the reciprocal ranks of all the good suggestions for all the queries, i.e.,

$$MRR = \frac{1}{q} \sum_{j=1}^{q} \frac{1}{Q_j} \sum_{i=1}^{Q_j} RR_{ji} \qquad (6)$$

where Q_j is the number of good suggestions for query j, q is the number of queries. For a query, its good query suggestions are determined partly by users' judgement and partly by the Google query suggestions in the experiment in this paper.

4.3.2 Mean average precision

MAP[29, 36] is an average precision across multiple queries and rankings. MAP assumes that users are interested in finding many relevant query suggestions and highly relevant suggestions should appear first in the list of suggestions.

Let the rank of the ith relevant query suggestion in the potential query suggestions made by a query suggestion method for query j be r_{ji}. The precision of the ith suggestion is defined as

$$P_{ji} = \frac{number\ of\ relevant\ suggestions}{number\ of\ suggestions\ examined} = \frac{i}{r_{ji}} \quad (7)$$

For an irrelevant suggestion, the precision is set to 0. MAP is defined as the average precision of all the query suggestions for all the queries, i.e.,

$$MAP = \frac{1}{q} \sum_{j=1}^{q} \frac{1}{Q_j} \sum_{i=1}^{Q_j} P_{ji} \qquad (8)$$

where Q_j is the number of relevant query suggestions for query j and q is the number of queries.

4.3.3 Discounted cumulated gain

MAP allows only binary relevance assessment (relevant or irrelevant), which does not distinguish highly relevant suggestions from mildly relevant suggestions. DCG[28, 29] is a metric that combines graded relevance assessments effectively.

Cumulative Gain (CG) of the Q_j query suggestions for query j is defined as

$$CG_j = w_1 + w_2 + \cdots w_{Q_j} \qquad (9)$$

where w_i is rating or weighting factor of the rank of the ith suggestion. Discounted Cumulative Gain (DCG) is defined by using a discount factor $1/(\log_2 i)$:

$$DCG_j = w_1 + \frac{w_2}{\log_2 2} + \frac{w_3}{\log_2 3} + \cdots \frac{w_{Q_j}}{\log_2 Q_j} \quad (10)$$

The average DCG (AvgDCG) over q queries is defined as

$$AvgDCG = \frac{1}{q} \sum_{j=1}^{q} DCG_j \qquad (11)$$

4.3.4 Precision at 10

Precision is defined here as the ratio of the number of relevant suggestions to the total number of irrelevant and relevant suggestions. This is a simple performance criterion and is often used as a baseline evaluation method.

Precision@10[36, 37] is the precision for the top ten query suggestions, which is calculated as follows:

$$P@10 = \frac{number\ of\ relevant\ suggestions\ among\ top\ 10}{10}$$

$$(12)$$

4.3.5 Integrated evaluation and user evaluation

The above four performance criteria emphasize different aspects of the performance. MRR is used to measure the performance of ranking, whilst P@10 is used to measure the performance of generating relevant query suggestions or

web documents. MAP and DCG can measure the performance of both ranking and producing relevant suggestions and web documents. Integrating the evaluation results from these four methods may lead to more comprehensive evaluation.

In order to check whether the evaluation using the above criteria is acceptable by real users, user evaluation will be conducted as well. Questionnaires as described in section 4.2 were used for this purpose.

4.4 Experimental results and evaluation

The experimental results are shown in the following tables, where an asterisk indicates that the related score differs significantly from the best one with the p value $< .05$. The method adopted for statistical significance test is t-test.

4.4.1 Evaluation of the quality of query suggestions

There are three methods which were evaluated and compared in this experiment: the combined method (Tfjac), TF-IDF (Tfidf), and Jaccard coefficient (Jac). By using four evaluation methods including integrated evaluation and user evaluation, the experimental results are illustrated below.

MRR results: The results of evaluation using MRR are given in Table 2, which show that the best query suggestion methods is Tfjac followed by Jac, and the ranking score of Tfidf is significantly lower than that of the best method.

Table 2: MRR results

QS methods	MRR scores	Rank
Tfidf	0.2934*	3*
Jac	0.3211	2
Tfjac	0.3846	1

MAP results: The results of evaluation using MAP are given in Table 3, which show that Tfjac is the best method for generating query suggestions in terms of ranking and producing relevant words.

Table 3: MAP results

QS methods	MAP scores	Rank
Tfidf	0.9544	2
Jac	0.9485	3
Tfjac	0.9712	1

P@10 results: The results of evaluation using P@10 are given in Table 4, which show that Jac and Tfjac have the same score and outperform Tfidf method in terms of generating relevant suggestions, and the score of Tfidf is significantly lower than those of the two best methods.

Table 4: P@10 results

QS methods	P@10 scores	Rank
Tfidf	0.9145*	3*
Jac	0.9524	1
Tfjac	0.9524	1

DCG results: The results of evaluation using DCG are given in Table 5, which show that Tfjac is the best method for ranking and producing highly relevant suggestions followed by Jac and Tfidf.

Table 5: DCG results

QS methods	DCG scores	Rank
Tfidf	8.0339	3
Jac	8.2880	2
Tfjac	8.5880	1

Integrated evaluation: Table 6 shows the rankings of the three query suggestion methods in terms of the four performance criteria respectively. For a method whose rank is significantly lower than the others, the rank is multiplied by two.

Table 6: Summary of evaluation results

QS methods	MRR ranking	MAP ranking	P@10 ranking	DCG ranking	Sum
Tfidf	3*(6)	2	3*(6)	3	17
Jac	2	3	1	2	8
Tfjac	1	1	1	1	4

The rankings in Table 6 can be transferred into MRR scores as shown in Table 7. It is clear that Tfjac is the best method overall for generating query suggestions, and Jac and Tfidf are significantly worse than Tfjac.

Table 7: Integrated evaluation in MRR scores

QS methods	MRR scores	Rank
Tfidf	0.2917*	3*
Jac	0.5833*	2*
Tfjac	1.0000	1

User evaluation: The results of the user rankings in MRR scores are given in Table 8, which show that the majority of the participants indicated that the query suggestions made by Tfjac were the best followed by Tfidf and Jac.

Table 8: User evaluation in MRR scores

QS methods	MRR scores	Rank
Tfidf	0.6495	2
Jac	0.6157	3
Tfjac	0.6732	1

These experimental results showed that Tfjac (a combined method based on TF-IDF and Jaccard coefficient method)

was the best method for query suggestion among the three methods evaluated in terms of all performance criteria including by user evaluation, significantly outperforming the most popularly used TF-IDF method.

4.4.2 *Evaluation of the effectiveness of the query suggestion method in interactive web search*

In the second experiment, the Tfjac method was used to generate query suggestions for interactive web search to evaluate the relevance of the top eight Google search returned documents by using query suggestions in comparison with that by using the original query only. The eighty test queries and the performance criteria used here are the same as in section 4.4.1, but the ranking is about the relevance of the returned web documents rather than directly the quality of query suggestions. The relevance ranks were obtained by eight participants as described in section 4.2.

MRR results: The results of evaluation using MRR are given in Table 9, which show that the documents returned by using the original query only were better ranked by the participants on average than by using query suggestion. However, there is no significant difference between the two MRR scores.

Table 9: MRR results

Methods	MRR scores
Query	0.4618
Query + suggestion	0.4452

MAP results: The results of evaluation using MAP are given in Table 10, which show that the documents returned by using query suggestion were significantly better than those by using the original query only in terms of the relevance of the returned documents to user's information need.

Table 10: MAP results

Methods	MAP scores
Query	0.9435
Query + suggestion	0.9740*

P@10 results: The results of evaluation using P@10 are given in Table 11, which show that the documents returned by using query suggestion had significantly higher precision scores. It means that these returned documents met users' information need significantly better.

Table 11: P@10 results

Methods	P@10 scores
Query	0.8422
Query + suggestion	0.9531*

DCG results: The results of evaluation using DCG are given in Table 12, which show that the DCG score obtained by using query suggestion is higher than that by using the original query only.

Table 12: DCG results

Methods	DCG scores
Query	9.1162
Query + suggestion	9.4021

In general, the second experiment demonstrated that the proposed query suggestion method is effective and it improves the relevance of the returned web documents through interactive web search.

5 Conclusions

This paper has investigated several ranked retrieval methods, adapted and combined them as well for query suggestion. Three query suggestion methods including the combined method developed in this paper have been evaluated using four performance criteria, integrated evaluation, and user evaluation as well. The experimental results show that Tfjac is the best for generating query suggestions among the three methods evaluated in terms of relevance and ranking. It is demonstrated that the Tfjac method is capable of combining the good query suggestions from both TF-IDF and Jaccard coefficient methods. The experimental results also indicate that the query suggestions made by the Tfjac method significantly improve the relevance of returned documents in interactive web search in terms of increasing the precision or the number of highly relevant documents. However, this combined method may deserve further investigation and there may be room for further improvement by using better combination strategies.

Performance evaluation usually depends on the queries used in the experiment and the judgment on the relevance of query suggestions with the original queries. This paper has designed eighty test queries related to eight topics based on Google search results and users' suggestions and adopted multiple performance criteria from different perspectives to ensure fair comparison and evaluation. However, further work may be conducted to overcome the limitation in this aspect of the performance evaluation and in the user evaluation conducted in this paper, for example, using standard or previously used benchmark datasets. Future work in line of this research may also include improving query suggestion by using knowledge base and user feedback, such as click-through data, through computational intelligence approaches.

References

[1] Fonseca BM, Golgher PB, de Moura ES, *et al*. Using association rules to discover search engines related queries, In: The First Latin American Web Congress, USA; 2003. p. 66-71.

[2] Delgado M, Martin-Bautista MJ, Sanchez D, *et al*. Association rules and fuzzy association rules to find new query terms. In: The Third Conference of the EUSFLAT, Lisbon, Portugal; 2009. p. 49-53.

[3] Kelly D, Cushing A, Dostert M, *et al*. Effects of popularity and quality on the usage of query suggestions during information search. In: CHI'2010, Atlanta, USA; 2010.

[4] Kato M, Saka, T, Tanaka K. Query session data *vs*. clickthrough data as query suggestion resources. In: ECIR; 2011. Dublin, Ireland; 2011.

[5] Carpineto C, Romano G. A Survey of automatic query expansion in information retrieval. ACM Computing Surveys. 2012; 44(1): 1-50. `http://dx.doi.org/10.1145/2071389.2071390`

[6] Niu X, Kelly D. The use of query suggestion during information search. Information Processing and Management: an International Journal. 2014; 50(1): 218-34. `http://dx.doi.org/10.1016/j.ipm.2013.09.002`

[7] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval: The Concepts and Technology behind Search. England: Pearson Education Limited; 2011.

[8] Nallapati R, Shah C. Evaluating the quality of query refinement suggestions in information retrieval. In: CIKM 2006, Arlington, Virginia, USA; 2006.

[9] Costa M, Miranda J, Cruz D, *et al*. Query suggestion for web archive search. In: The 10th International Conference on Preservation of Digital Objects (iPres 2013), Lisbon, Portugal; 2013.

[10] Kato M, Sakai T, Tanaka K. Structured query suggestion for specialization and parallel movement: effect on search behaviours. In: WWW 2012, Lyon, France; 2012. p. 389-98.

[11] Baeza-Yates R, Hurtado C, Mendoza M. Query recommendation using query logs in search engines. In: The 2004 International Conference on Current Trends in Database Technology; 2004. p. 588-96.

[12] Boldi P, Bonchi F, Castillo C, *et al*. Query suggestion using query flow graphs. In: The 2009 Workshop on Web Search Click Data, Milan, Italy; 2009. p. 56-63.

[13] Boldi P, Bonchi F, Castillo C, *et al*. The query flow graph: model and applications. In: CIKM'08, California, USA; 2008.

[14] Cao H, Jiang D, Pei J, *et al*. Context-aware query suggestion by mining click-through and session data. In: KDD'2008, Nevada, USA; 2008.

[15] Huang C, Chien L, Oyang Y. Relevant term suggestion in interactive web search based on contextual information in query session logs. Journal of the American Society for Information Science and Technology. 2003; 54(7): 638-49. `http://dx.doi.org/10.1002/asi.10256`

[16] Kruschwitz U, Lungley D, Albakour M, *et al*. Deriving query suggestions for site search. Journal of the American Society for Information Science and Technology. 2013; 64(10): 1975–94. `http://dx.doi.org/10.1002/asi.22901`

[17] Liao Z, Song Y, Huang Y, *et al*. Task trail: an effective segmentation of user search behavior. IEEE Transactions on Knowledge and Data Engineering (in press); 2014.

[18] Mei Q, Zhou D, Church K. Query suggestion using hitting time. In: CIKM'08, California, USA; 2008.

[19] Gong Z, Cheang C, Hou L. Web query expansion by WordNet. LNCS. 2005; 3588: 166-75.

[20] Wan J, Wang W, Yi J, *et al*. Query expansion approach based on ontology and local context analysis. Research Journal of Applied Sciences, Engineering and Technology. 2012: 4(16): 2839-43.

[21] Hu H, Zhang M, He Z, *et al*. Diversifying query suggestions by using topics from Wikipedia. In: The 2013 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Atlanta, USA; 2013.

[22] Biega J, Kuzey E, Suchanek F. Inside YAGO2s: A transparent information extraction architecture. In: WWW 2013, Rio de Janeiro, Brazil; 2013.

[23] Hoffart J, Suchanek F, Berberich K, *et al*. YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In: WWW 2011, Hyderabad, India; 2011.

[24] Suchanek F, Kasneci G, Weikum G. YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In: WWW 2007, Banff, Alberta, Canada; 2007.

[25] Suchanek F, Hoffart J, Kuzey E, *et al*. YAGO2s: modular high-quality information extraction with an application to flight planning. In: The German Computer Science Symposium (BTW 2013), Magdeburg, Germany; 2013.

[26] Yang J, Cai R, Jing F, *et al*. Search-based query suggestion. In: CIKM'08, California, USA; 2008.

[27] Song Y, Zhou D, He L. Query suggestion by constructing term-transition graphs. In: WSDM'12, Seattle, Washington, USA; 2012.

[28] Jurafsky D, Martin JH. Speech and Language Processing: An Introduction to Natural Language Processing. Computational Linguistics and Speech Recognition. Second Edition, Prentice Hall; 2008.

[29] Manning CD, Raghavan P, Schutze H. Introduction to Information Retrieval. England: Cambridge University Press; 2008.

[30] Yih W, Qazvinian V. Measuring word relatedness using heterogeneous vector space models. In: The 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2012), Montreal, Canada; 2012.

[31] Kulkarni S, Caragea D. Computation of the semantic relatedness between words using concept clouds. In: KDIR 2009, Madeira, Portugal; 2009. p. 183-8.

[32] Blizard WD. Multiset theory. Notre Dame Journal of Formal Logic. 1989; 30(1): 36-66.

[33] He B, Ounis I. Studying query expansion effectiveness. In: The 31th European Conference on IR Research on Advances in Information Retrieval, Toulouse, France; 2009. p. 611-9.

[34] Bordag S. A comparison of co-occurrence and similarity measures as simulations of context. In: The 9th International Conference on Computational Linguistics and Intelligent Text Processing, Springer-Verlag Berlin, Heidelberg; 2008. p. 52-63.

[35] Dybkjaer L, Hemsen H, Minker W. (Eds.). Evaluation of Text and Speech Systems. Springer, Dordrecht, Netherlands; 2007.

[36] Otegi A, Arregi X, Agirre E. Query expansion for IR using knowledge-based relatedness. In: The 5th International Joint Conference on NLP, Chang Mai, Thailand; 2011. p. 1467-71.

[37] Okabe M, Yamada S. Semisupervised query expansion with minimal feedback. IEEE Transactions on Knowledge and Data Engineering. 2007; 19(11): 1585-9. `http://dx.doi.org/10.1109/TKDE.2007.190646`