HAL open science

New technique to deal with verbose queries in social book search

Messaoud Chaa, Omar Nouali, Patrice Bellot

To cite this version:

Messaoud Chaa, Omar Nouali, Patrice Bellot. New technique to deal with verbose queries in social book search. 2017 ACM/IEEE/WIC International Conference on Web Intelligence, Aug 2017, Leipzig, Germany. pp.799-806, 10.1145/3106426.3106481 . hal-01794695

HAL Id: hal-01794695

https://hal.science/hal-01794695v1

Submitted on 25 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

New Technique to Deal With Verbose Queries in Social Book Search

Messaoud Chaa
Research Center on Scientific and Technical Information
Université Abderrahmane Mira Béjaia
Béjaia, Algeria, 06000
mchaa@cerist.dz

Omar Nouali
Research Center on Scientific and Technical Information
onouali@cerist.dz

Patrice Bellot
Aix-Marseille Université, CNRS, LSIS
Marseille, France
patrice.bellot@lsis.org

ABSTRACT

Verbose query reduction and query term weighting are automatic techniques to deal with verbose queries. The objective is either to assign an appropriate weight to query terms according to their importance in the topic, or outright remove unsuitable terms from the query and keep only the suitable terms to the topic and user's need. These techniques improve performance and provide good results for ad hoc information retrieval. In this paper we propose a new approach to deal with long verbose queries in Social Information Retrieval (SIR) by taking Social Book Search as an example. In this approach, a new statistical measure was introduced to reduce and weight terms of verbose queries. Next, we expand the query by exploiting the similar books mentioned by users in their queries. We find that the proposed approach improves significantly the results.

CCS CONCEPTS

• **Information systems → Information retrieval query processing**; Information retrieval; **Query representation**;

KEYWORDS

Verbose Query Reduction, Query Term Weighting, Quey Expansion, Tf.Iqf, Social Book Search, Stop-Word List

ACM Reference format:
Messaoud Chaa, Omar Nouali, and Patrice Bellot. 2017. New Technique to Deal With Verbose Queries in Social Book Search. In *Proceedings of WI '17, Leipzig, Germany, August 23-26, 2017,* 8 pages.
https://doi.org/10.1145/3106426.3106481

1 INTRODUCTION

It is certainly that the most of queries, submitted to search engine, are less than five terms long [11]. However, it is more advantageous, for web users, to use natural language to express their needs in detail instead of using minimal keywords in their queries. Thus, the use of long verbose queries by users has increased over time and the average query length has grown year after year[28]. Moreover, the emergence of several applications such as question answering systems[1] , discussion forum like LibraryThing[2] and voice queries on mobile devices[3]. Such applications allow users to utilize natural language to express their information needs in the form of long and verbose queries.

On the other hand, and as most of information retrieval models [7, 24, 25] are based on the term frequency of all query terms, to retrieve and rank relevant documents to the query. The performance of such models, with long queries, is not always satisfactory. This is due that most of retrieval models assume that all terms given by user in verbose natural language query are equally important and tend to retrieve documents that contain all of the query terms, while this is not the case. Some terms are completely extraneous to the context of the request and they have been used only as: an introduction to the request "Hey there! I am looking for suggestions/ recommendations for reading about...", or as a conclusion to the request "Thanks for any help.", "Any suggestions!".

The study of [4] has shown that search engines generally perform poorly on verbose natural language queries when compared to short keyword queries. This comparison was made using the queries of TREC corpora which considers that short queries are those with four terms at the most and long queries are those containing five terms or more. The same study indicates that the longer is the query the less efficient is the search engine. Moreover, a recent empirical study of [9] indicates that 73% of verbose queries, used in TR-based software maintenance are improved when the number of the query terms are reduced. The results increased by 21.8% and 13.4% in terms of MRR and MAP respectively. To tackle the problem of verbosity in natural language queries and improve search engine effectiveness, verbose queries have received more attention in recent years[4, 10, 23]. Most of them are classified in two main categories: Query Reduction and Query term weighting Approaches. Instead of using all query terms of the original query, the query reduction approaches select only a subset of important terms to form the reduced query and submits them to search engine. The Query term weighting Approaches consist of assigning to each term, an appropriate weight, before submitting the new query to the search engine.

In summary, all this works, whether from query reduction approaches or query term weighting approaches, show an improvement in search engine performance over verbose long queries. Despite the fact that most of these works have focused on Ad-hoc information retrieval, there exist others that tackled specific domains:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WI '17, August 23-26, 2017, Leipzig, Germany
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4951-2/17/08…$15.00
https://doi.org/10.1145/3106426.3106481

[1] https://uk.answers.yahoo.com/
[2] http://www.librarything.com
[3] Google Voice Search

web image retrieval[12], e-commerce[32], spoken queries[1, 20], and medical document retrieval[3]. And to our knowledge no study has focused on social information retrieval[8].

In this paper, we have decided to handle verbose queries for social information retrieval (SIR) using social book search (SBS) collection as test to evaluate our approach. In this study, our target is to reduce the verbose queries by removing inappropriate terms (Stopwords) and keep only the appropriate terms to the topic and user's need. Then we weight all terms of the query according their importance by using a new proposed statistical measure (tf.iqf). Finally we expand the queries by adding terms from the similar books mentioned by users.

The remainder of the paper is as follows: Section 2 outlines related works in which we present some recent works in query reduction and query term weighting. Section 3 explains the main goal of social book search as well as the test collection and topics used. Section 4 is to describe our proposed approach and explain the three techniques proposed to deal with verbose queries. In section 5 we describe our experimental results, discuss and analyze the results in section 6. Finally, Section 7 provides a conclusion.

2 RELATED WORKS

As mentioned above, works on processing verbose queries are categorized in two main categories. In this section we introduce in details some related works for each category.

2.1 Query term weighting Approaches

Based on the idea that in natural language queries, each term can have different weights in different queries depending on the context of the query. Therefore, assigning a weight to query terms should have a significant impact on the results returned by search engine. [4] used machine learning technique for identification and weighting query concepts. This method was achieved by representing each concept by different features and significantly improves retrieval effectiveness using a large set of natural language queries derived from TREC topics. Another work of [19] propose a learning to rank framework to weight all term of the query instead of concept. [31] used Hidden markov Model to weight terms in vebose queries. In this work, Part-of-speech (POS) features of terms are used as observations and the weight levels of the query terms as the hidden states. Drawing on an idea from text summarization, [23] propose an unsupervised method to estimate which term are most central to the query. An initial set of more relevant documents to the original query are used to define a recursion on the query word weight vector that converges to a fixed point representing the vector that optimally describes the initial result set. Recently,[10] propose a discriminative query language modeling by estimating the probability that a particular query term is topical and correctly weights the salient aspects of the query. Since the approach is based on language model, the frequency of terms in the query, in collection, across all queries and the length of the query are used as features.

2.2 Query Reduction Approaches

Based on the idea that natural language verbose queries contain many noisy terms, i.e., terms that are extraneous to the context of the request and might cause a topic drift. Therefore, query reduction whom remove inappropriate terms from long queries and keep only the appropriate terms to the topic and user's need should improve the performance of search engine. [18] proposed to generate a set of sub-queries (subset of terms) from the original query and then using RankSVM [14] to rank sub-queries. Finally, the top-ranked sub-query was selected to replace the original query. The work shows an 8% significant average improvement of the mean average precision. In the same context, [30] rank sub-queries using Conditional Random Field model (CRF) however they select the top-ranked sub-queries instead of the best sub-query, to replace the original query, using several types of retrieval models. [29] proposed a reformulation tree framework to organize the sub-queries as a tree structure, where each node is a reformulated query (sub-query). Considering the relationships between nodes, a weight estimation approach assigns weights to each sub-query and directly optimizing the retrieval performance. [32] build a classifier to predict which term is the most likely to be deleted from a given query using various term-dependent and query-dependent measures as features. The authors validate their approach using a large collection of query sessions logs from an e-commerce site.

```
<book>
<isbn>0194518000</isbn>
<title>New English File: Student's Book Intermediate level</title>
<listprice>$26.61</listprice>
<publisher>Oxford University Press</publisher>
<reviews>
<review>
<date>2006-07-20</date>
<summary>New English File Elementary</summary>
<content>New English File, as its name suggests, is the new and improved version of English File. Unlike the original English File, which came in four levels (Beginner, Pre-Intermediate, Intermediate and Upper-Intermediate),...Thank you for your attention. </content>
<rating>5</rating>
<totalvotes>4</totalvotes>
<helpfulvotes>4</helpfulvotes>
</review>
</reviews>
<tags>
<tag count="1">english</tag>
<tag count="1">hard</tag>
<tag count="1">en</tag>
<tag count="1">@home-a</tag>
</tags>
</book>
```

Figure 1: XML file represent an example book in Social Book Search.

Table 1: Top ranked terms with the corresponding number of queries in which they appear.

Stem of term	#topics	Stem of term	#topics
The	966	But	534
And	895	About	438
Book	816	With	432
For	803	Thi	420
Read	627	Anyon	417
That	600	Suggest	416
Ani	600	Look	414
Recommend	571	Can	381
Have	567	Like	375

3 SOCIAL BOOK SEARCH

The main goal of Social Book Search is to evaluate the impact of professional metadata as well as user-generated social content (e.g., reviews, tags and ratings) for book search on the Web by developing new techniques and tools to support users to find the most relevant books to their needs [16]. To reach this goal and since 2011, INEX SBS has provided a document collection of 2.8 million books containing both professional and social metadata (see Fig. 800). Professional metadata (e.g. ISBN, title, author, publisher...) and social metadata (reviews and rating) were extracted from Amazon whereas LibraryThing (LT) was utilized to extract tags assigned to books by users. In order to evaluate systems in SBS, a set of topics have been also made available. These topics are based on discussion threads from LT forums and contain many fields namely: group, title, narrative and examples (see Fig. 802). The group field means the discussion group in which the user posts their thread. The title is often a brief summary of the user' information need but sometimes out of the context, while in narrative the user utilizes natural language to explain their needs in details. As of examples field, it consists of a small number of similar books to the request that some LibraryThing users provide in their topics in order to indicate the kind of books they request. The terms of those example books can be used to expand the original query considering that those examples are pseudo-relevant to the query. From the narrative field we realize that the queries in SBS are verbose and sometimes contain much more than just the information need. This extraneous information to user need can lead to topic drift. For this reason, we decided remove this extraneous terms and keep only the appropriate to the context of the request and the users' information need.

4 PROPOSED APPROACH

Our proposed approach based on the following key intuitions:

The more frequently a term appears in many queries, the less informative it is whilst the more times this term appears in the same query, the more relevant this term is to the query context.

Terms present in the set of similar books mentioned by users in their queries could be relevant to the query context.

According to the two intuitions mentioned above, our approch includes three techniques: (i) stopword removal technique to reduce the verbose queries, (ii) new statistical measure tf.iqf to weighting terms and (iii) the expansion of this query using similar books mentioned in the topic.

4.1 Stopword Removal for Query Reduction

The work of [21] shows that constructing a specific stopword list of a given collection can improve the performance results. In their works, all terms of the collection are ranked using according to their importance in the collection. Then they choose a threshold and any words that appear above the particular threshold are treated as stop-words and will not be indexed. In the same way, [13] proposed to chose to apply these technique by removing from the query all words which occur on the stopword list. Both works utilize the statistic measure IDF to rank terms and decide which term is a stopword or not. Unlike the approaches discussed above, which take into account the number of documents in which the term appears. Our approach uses the number of queries in which the term appears instead. First, we ranked the terms according to the number of times they appear in the queries. Then we choose a threshold and all the terms that appear above will construct a stopwords list. Finally the terms of the list will be removed from each query in order to form a reduced query (RQ) before querying by an IR model. Table 1 shows a sample of the top ranked terms with the corresponding number of queries in which they appear.

4.2 Query terms weighting

The frequency of terms in the query, in the corpus or in the external resources are commonly used by researchers to weight terms on verbose queries [15, 30]. In our case, and based on the first intuition mentioned in section 4 and inspired from the commonly used term weighting tf.idf[27], we have introduced a new measure Term Frequency-Inverse Query Frequency (tf.iqf). This measure was used in order to increase the weight of terms which appear the less in the set of queries and decrease the weight of terms which appear the most in the set of queries. The tf.iqf measure is calculated as follow:

$$TFIQF(t) = tf(t,q).iqf(t) \tag{1}$$

Where *tf(t,q)* is the frequency of term *t* in the topic *q*, and the *iqf(t)* is the inverse query frequency calculated as follow:

$$iqf(t) = log\frac{|Q| - qf(t) + 0.5}{qf(t) + 0.5} \tag{2}$$

Where qf(t) is the number of topics that contain t, and |Q| is the total number of topics in a collection (all topics from INEX 2011 to 2016 are used).

4.3 Query expansion

At first glance,the expansion of verbose queries seems counterintuitive. Nevertheless,query expansion has improved performance in such queries. Various query expansion techniques have been developped like adding a category labels to long queries [2]], adding latent concepts extracted from pseudo-relevance feedback [5] or using multiple information sources [6]]and interactive query expansion using pseudo-relevance feedback [17] In our case, and assuming that similar books mentioned by users in their topics are relevant, their terms are also important to the query. In order to

```
<topicid>107277</topicid>
<group>FantasyFans</group>
<title>Fantasy books with creative heroines?</title>
<request>Greetings! I'm looking for suggestions of fantasy novels whose heroines are creative in some way and have some sort of
talent in art, music, or literature. I've seen my share of "tough gals" who know how to swing a sword or throw a punch but have next
to nothing in the way of imagination. I'd like to see a few fantasy-genre Anne Shirleys or Jo Marches. Juliet Marillier is one of my
favorite authors because she makes a point of giving most of her heroines creative talents. Even her most "ordinary" heroines have
imagination and use it to create. Clodagh from "Heir to Sevenwaters," for example, may see herself as being purely domestic, but she
plays the harp and can even compose songs and stories. Creidhe of "Foxmask" can't read, but she can weave stories and make colors.
The less ordinary heroines, like Sorcha from "Daughter of the Forest" and Liadan from "Son of the Shadows," are good storytellers. I'm
looking for more heroines like these. Any suggestions? </request>
<examples>
<example> <booktitle>Daughter of the Forest</booktitle> <author>Juliet Marillier</author>
<workid>6442</workid>
</example>
<example><booktitle>Foxmask</booktitle> <author>Juliet Marillier</author>
<workid>349475</workid>
</example>
<example> <booktitle>Son of the Shadows</booktitle> <author>Juliet Marillier</author>
<workid>6471</workid>
</example>
<example> <booktitle>Heir to Sevenwaters</booktitle> <author>Juliet Marillier</author>
<workid>5161003</workid>
</example>
</examples>
<catalogue/>
```


Figure 2: XML file represent an example topic in Social Book Search.

Table 2: Details of the six years topics used for the experiment

Year	#Topics	Fields
2011	211	Title,Group,Narrative,type,genre,specificity
2012	96	Title,Group,Narrative,type,genre
2013	370	Title,Group,Narrative,Query
2014	672	Title,Group,Narrative,mediated_query
2015	178	Title,Group,Narrative,mediated_query
2016	119	Title,Group,Request

exploit these similar books, we expand the weighted reduced query (WRQ) by automatically adding terms from these similar books. Rocchio relevance feedback [26] is one of the most popular techniques used for this task. The function used to expand the queries is as follow :

$$\overrightarrow{EWRQ} = \overrightarrow{WRQ} + \frac{\beta}{|S|} \sum_{d \in S} \overrightarrow{d} \tag{3}$$

Where $\overrightarrow{EWRQ}$ and $\overrightarrow{WRQ}$ are the expanded wheighted reduced query and the weighted reduced query vector respectively, $\overrightarrow{d}$ denotes the weighted term vector of the similar book *d*. *S* is the set of similar books and |S| is the number of similar books.

5 EXPERIMENTAL RESULTS

As mentioned-above the document collection used in our experiments has a total of 2.8 million of records extracted from Amazon and Librarything Forum. For each book the professional metadata was ignored and only the social information data (tags assigned to books by users in LT Forum) were used to represent books. As regards topics, from 2011 to 2016 the organizers of SBS have used Librarything forum to extract a different set of topics with relevance judgments for each year. Table 2 summarizes the set of topics utilized for each year with some details. The same table shows that the majority of topics contain the three fields (title, group and narrative), while the 2011 and 2012 topics contain (type, genre and specificity) which are ignored in our experiments. In 2013 and 2014, the field entitled query (2013) and mediated query (2014) are just the same and are created manually by a trained annotator. This field is provided by the organizers of SBS to compensate for non-representative thread titles for some of the forum topics. The example field which contains a list of books that are related to the topic is present in all years. The Terrier IR platform [22] was used to index the collection by applying basic stopword filtering and Porter stemming algorithm. The BM25 model was used for querying with the parameters (b=0, k3=1000, k1=2). Using the BM25 model, the relevance score of a book d for query Q is given by:

$$S(d,Q) = \sum_{t \in Q} \frac{(k_1+1)w(t,d)}{k_1 + w(t,d)} . idf(t) . \frac{(k_3+1)w(t,Q)}{k_3 + w(t,Q)} \tag{4}$$

Table 3: results in term of ndcg@10 obtained for different values of the thresholds for query reduction.

Threshold	5	10	15	20	25	30	35	40	45	50	55	60
2011	0.2140	0.2308	0.2376	**0.2421**	0.2385	0.2362	0.2330	0.2367	0.2344	0.2291	0.2222	0.2173
2012	0.1734	0.1777	0.1830	**0.1960**	0.1930	0.1921	0.1935	0.1940	0.1947	0.1898	0.1870	0.1818
2013	0.1102	0.1277	0.1325	0.1299	0.1327	**0.1342**	0.1296	0.1266	0.1247	0.1261	0.1260	0.1255
2014	0.1060	0.1179	0.1209	0.1212	0.1218	**0.1257**	0.1181	0.1171	0.1133	0.1129	0.1128	0.1108
2015	0.0856	0.0885	0.0944	0.0937	0.0933	**0.0997**	0.0935	0.0934	0.094	0.0891	0.0873	0.0892
2016	0.1198	0.1283	0.1298	**0.1338**	0.1286	0.1264	0.1259	0.1250	0.1227	0.1200	0.1125	0.1117
All	0.1235	0.1356	0.1399	0.1410	0.1408	**0.1429**	0.1377	0.1371	0.1347	0.1332	0.1313	0.1296

Table 4: Comparison in term of ndcg@10 of the three representations of the query.

Year	Title only	Narrative only	Title+Narrative
2011	**0.2567**	0.0715	0.1170
2012	**0.1804**	0.0526	0.1025
2013	**0.1158**	0.0483	0.0719
2014	**0.1167**	0.0506	0.0717
2015	**0.1164**	0.0442	0.0615
2016	**0.1145**	0.0370	0.0595
ALL	**0.1380**	0.0512	0.0774

Table 5: Comparison in terms of ndcg@10 of the combination of the three techniques (weigthing, reducing and expansion).

Year	Baseline model	RQ	WRQ	EWRQ
2011	0.2567	0.2362	0.2675	**0.3291**
2012	0.1804	0.1921	0.2036	**0.2210**
2013	0.1158	0.1342	0.1444	**0.1524**
2014	0.1167	0.1257	0.1336	**0.1565**
2015	0.1164	0.0997	0.1176	**0.1380**
2016	0.1145	0.1264	0.1411	**0.1500**
ALL	0.1380	0.1429	0.1561	**0.1790**

Where w(t,d) and w(t,Q) are respectively the weigths of terms in document d and in query Q. idf(t) is the inverse document frequency of term t, given as follow:

$$idf(t) = log\frac{|D| - df(t) + 0.5}{df(t) + 0.5} \quad (5)$$

Where *df(t)* is the number of documents tagged with *t*, and |D| is the number of documents in the collection.

In the first step of our experiments we build a three set of queries according to the topic field used as a query namely: title only, narrative only and title+narrative. The results are presented in table 4. The table shows the modest results obtained from narrative representation of the query. These were caused by the verbosity of the narrative. However the user used it to better explain their need which means that it contains some informative terms. Thus, we combined the title with the narrative as a representation of the queries and we generated a stop-words list to reduce these queries. To do so, we varied the threshold values (number of queries in which the term appears), in order to find one particular set of stop-words list that would produce a better ndcg@10. The threshold was varied from 5 to 60, with 5 steps. After several experiments on all six years topics, Table 3 shows the results obtained. From this table, we notice that, whatever the value of the threshold, query reduction (after removing stop words from queries) performs better than the case of no reduction (column title+narrative in table 4). However, the same table indicates that most of these different sets of topics achieve the optimal results when the threshold is set to 20 or 30. The model achieves the optimal results across all queries in term of ndcg@10 (0.1429) when the threshold is set to 30. Then we chose setting the threshold to 30 for the remaining experiments. This means that each term that appears in more than thirty topics is considered as a stop word and must be removed from queries.

Secondly, we used the tf.iqf function (2) in section 4.2 to weight terms in the reduced query. The result of this step gave a new reduced query with weighted terms. Finally, the weighted reduced query was expanded by adding new terms from similar books using the function (4) in section 4.3. The rocchio function was used with their default parameter settings β=0.4, and the number of terms selected from each similar book was set to 10. Table 5 presents the results of the whole process. Where the columns RQ, WRQ and EWRQ represent the results obtained by the reduced query, weighted reduced query and expanded weighted reduced query respectively.

The different representations as well as the results obtained for the topic 107277 in term of ndcg@10 and map are shown in Table 6. This is a good example to show the positive impact of our approach on the final results. We can see that when we use title, narrative or title+narrative the ndcg@10=0 whilst the use of our approach(RQ, WRQ, EWRQ) gradually increases the ndcg@10 to 0.2148, 0.4714 and 0.5959. As to map, there is significant improvement of the results when applying our approach compared to the intial query submitted by the user.

6 RESULTS AND ANALYSIS

In this section, we present the findings of our experiments across all six years topics. Table 4 shows that using only the title field of topics as a query gives better results comparing to when using narrative only or the combination of title and narrative. This reinforces what has been already stated namely that search engines generally perform poorly on verbose queries when compared to short keyword

Table 6: The diffrent representations of the topic 107277 obtained from the deffrent techniques used in our approch as well as the results obtained.

representation	query	ndcg@10	map
title	fantasi :1 book :1 creativ :1 heroin :1	0.0000	0.0233
Narrative	heroin :1 like :0.6 can :0.6 look :0.4 suggest :0.4 fantasi :0.4 creativ :0.4 wai :0.4 talent :0.4 imagin :0.4 see :0.4 ordinari :0.4 stori :0.4 greet :0.2 novel :0.2 sort :0.2 art :0.2 music :0.2 literatur :0.2 ve :0.2 seen :0.2 share :0.2 tough :0.2 gal :0.2 who :0.2 know :0.2 swing :0.2 sword :0.2 throw :0.2 punch :0.2 next :0.2 genr :0.2 ann :0.2 shirlei :0.2 jo :0.2 march :0.2 juliet :0.2 marilli :0.2 favorit :0.2 author :0.2 make :0.2 point :0.2 give :0.2 creat :0.2 clodagh :0.2 heir :0.2 sevenwat :0.2 exampl :0.2 mai :0.2 be :0.2 pure :0.2 domest :0.2 plai :0.2 harp :0.2 compos :0.2 song :0.2 creidh :0.2 foxmask :0.2 read :0.2 weav :0.2 color :0.2 sorcha :0.2 daughter :0.2 forest :0.2 liadan :0.2 son :0.2 shadow :0.2 storytel :0.2	0.0000	0.0083
title+narrative	heroin :1 fantasi :0.5 creativ :0.5 like :0.5 can :0.5 look :0.33 suggest :0.33 wai :0.33 talent :0.33 imagin :0.33 see :0.33 ordinari :0.33 stori :0.33 book :0.17 greet :0.17 novel :0.17 sort :0.17 art :0.17 music :0.17 literatur :0.17 ve :0.17 seen :0.17 share :0.17 tough :0.17 gal :0.17 who :0.17 know :0.17 swing :0.17 sword :0.17 throw :0.17 punch :0.17 next :0.17 genr :0.17 ann :0.17 shirlei :0.17 jo :0.17 march :0.17 juliet :0.17 marilli :0.17 favorit :0.17 author :0.17 make :0.17 point :0.17 give :0.17 creat :0.17 clodagh :0.17 heir :0.17 sevenwat :0.17 exampl :0.17 mai :0.17 be :0.17 pure :0.17 domest :0.17 plai :0.17 harp :0.17 compos :0.17 song :0.17 creidh :0.17 foxmask :0.17 read :0.17 weav :0.17 color :0.17 sorcha :0.17 daughter :0.17 forest :0.17 liadan :0.17 son :0.17 shadow :0.17 storytel :0.17	0.0000	0.0100
RQ	heroin :1 creativ :0.5 talent :0.33 imagin :0.33 ordinari :0.33 greet :0.17 art :0.17 music :0.17 seen :0.17 share :0.17 tough :0.17 gal :0.17 swing :0.17 sword :0.17 throw :0.17 punch :0.17 ann :0.17 shirlei :0.17 jo :0.17 march :0.17 juliet :0.17 marilli :0.17 creat :0.17 clodagh :0.17 heir :0.17 sevenwat :0.17 pure :0.17 domest :0.17 plai :0.17 harp :0.17 compos :0.17 song :0.17 creidh :0.17 foxmask :0.17 weav :0.17 color :0.17 sorcha :0.17 daughter :0.17 forest :0.17 liadan :0.17 son :0.17 shadow :0.17 storytel :0.17	0.2148	0.0276
WRQ	heroin :1 creativ :0.6 talent :0.43 ordinari :0.42 imagin :0.32 marilli :0.26 creidh :0.26 clodagh :0.26 weav :0.26 domest :0.26 harp :0.26 liadan :0.26 sorcha :0.26 compos :0.26 sevenwat :0.26 foxmask :0.26 storytel :0.24 pure :0.24 swing :0.24 gal :0.24 punch :0.24 march :0.24 juliet :0.24 shirlei :0.23 heir :0.21 throw :0.21 forest :0.21 shadow :0.21 greet :0.21 sword :0.21 song :0.2 plai :0.18 tough :0.18 color :0.18 music :0.17 creat :0.16 son :0.16 ann :0.16 seen :0.16 art :0.15 daughter :0.15 share :0.15	0.4714	0.0575
EWRQ	sevenwat :1.26 heroin :1 fantasi :0.75 creativ :0.6 marilli :0.5 talent :0.43 ordinari :0.41 juliet :0.4 celtic :0.38 ireland :0.38 fairi :0.37 swan :0.36 tale :0.34 imagin :0.32 creidh :0.26 clodagh :0.26 weav :0.26 domest :0.26 harp :0.26 liadan :0.26 sorcha :0.26 compos :0.26 foxmask :0.26 storytel :0.24 pure :0.24 swing :0.24 gal :0.24 punch :0.24 march :0.24 trilogi :0.23 shirlei :0.23 heir :0.21 throw :0.21 forest :0.21 shadow :0.21 greet :0.21 sword :0.21 song :0.2 histor :0.19 plai :0.18 tough :0.18 color :0.18 music :0.17 seri :0.17 creat :0.16 son :0.16 ann :0.16 retel :0.16 seen :0.16 art :0.15 daughter :0.15 share :0.15 mytholog :0.14 read :0.11 magic :0.11 fairytal :0.1 romanc :0.1 retold :0.1 fiction :0.09 folklor :0.07 fae :0.07 love :0.07 palencar :0.07 druid :0.06 faeri :0.06 folk :0.06 irish :0.06 myth :0.06 romant :0.06 femal :0.06 tbr :0.06 strong :0.06 australian :0.05 wild :0.05 specul :0.05 keltisch :0.05 jude :0.05 unread :0.05 adventur :0.05 fi :0.05 sci :0.05	**0.5959**	**0.0907**

queries. We consider the results obtained by the title field of the topic as a baseline model. We performed the three reformulation of the query (reduced query, reduced weighted query and expanded weighted reduced query). Table 5 shows the results obtained by our approach for the three reformulations of the query. For the query reduction we show that this technique outperforms the baseline in only four of the six sets of topics. However, for all queries the ndcg@10 increases from 0.1380 to 0.1429. From the same table we show that when applying both weighing function tf.iqf technique as well as the query expansion technique the results are better across all the sets of queries. In term of ndcg@10 the results increase from 0.1429 to 0.1561 when applying the term weighting technique and from 0.1561 to 0.1790 when using the query expansion technique.

We further compare the performance of our best results obtained by (EWRQ)to the best official runs of the six years in Social Book Search. Table 7 represent the comparative results. The results show that the ndcg@10 value of our best results is better than the best runs of the four years (2011, 2012, 2013 and 2014) but lower than the best runs of the two years(2015 and 2016). The table shows also that our approach outperforms the best runs across all topics ndcg@10=0.1790 of our approach against ndcg@10=0.1726 for the best runs. As to map, we can seen that our approach outperforms

Table 7: The comparison in terms of ndcg@10 and map of our best results (EWRQ) to the official best runs submitted to Social Book Search across all six years

	Our approach		Best run	
	ndcg@10	map	ndcg@10	map
2011	**0.3291**	0.1648	0.3101	**0.2283**
2012	**0.2210**	**0.1975**	0.1456	0.1249
2013	**0.1524**	**0.114**	0.1361	0.0861
2014	**0.1565**	**0.1100**	0.1420	0.1020
2015	0.1380	0.0843	**0.1870**	**0.1050**
2016	0.1500	0.0880	**0.2157**	**0.1253**
All	**0.1790**	**0.1187**	0.1730	0.1180

Table 8: The p-value obtained using the Statistical Significance (two-tailed T-Test), figures in bold indicate a significance of the difference between two methods at level 0.05

Method1	Method2	p-value
Baseline Model	RQ	0.2364
Baseline Model	WRQ	**0.0038**
Baseline Model	EWRQ	**0.000002**
RQ	WRQ	0.0896
RQ	EWRQ	**0.0003**
WRQ	EWRQ	0.0601

the best runs only in three years (2012, 2013 and 2014). but for all topics, our approch outperforms slightly the best runs.

In order to report more informatively the effect of reduction, weighting and expansion techniques, the statistical significant testing (two-tailed t-test) has been conducted. As shown in table 8 the reducing technique (RQ) is not statistically significant compared to the baseline (p-value = 0.23) , this due that in the reduced query we used the title and narrative while in the base line only the title were used. From the same table we show that combining reducing and weighting techniques (WRQ) is statistically significant compared to baseline (p-value=0.0038) but less significant compared to RQ. Combining all techniques (EWRQ) is statistically significant compared to the baseline (p-value = 0.000002) or to reducing technique (p-value=0.003) , but less significant compared to WRQ (p-value = 0.0601), this could be due that similar books does not present in all topics and only the topics that contains similar books have had improvements.

7 CONCLUSIONS

In this paper, we proposed a new technique to automatically generate a stopword list in order to reduce verbose queries. Moreover, we introduced a new function, called tf.iqf, inspired by the tfidf measure, to weight terms and measure how informative a given term is. In addition, the rocchio technique was used to add new terms from the similar books mentioned by users in their topics. We investigated the effectiveness and the robustness of this new approach using the social book search collection. The experimental results demonstrate that the proposed approach has given satisfactory results with all the three techniques that have been used.

REFERENCES

[1] Jaime Arguello, Sandeep Avula, and Fernando Diaz. 2017. Using Query Performance Predictors to Reduce Spoken Queries. In *European Conference on Information Retrieval*. Springer, 27–39.

[2] Peter Bailey, Ryen W White, Han Liu, and Giridhar Kumaran. 2010. Mining historic query trails to label long and rare search engine queries. *ACM Transactions on the Web (TWEB)* 4, 4 (2010), 15.

[3] Saeid Balaneshin-kordan and Alexander Kotov. 2016. Optimization Method for Weighting Explicit and Latent Concepts in Clinical Decision Support Queries. (2016). https://doi.org/10.1145/2970398.2970418

[4] Michael Bendersky and W Bruce Croft. 2008. Discovering key concepts in verbose queries. *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (2008), 491–498. https://doi.org/10.1145/1390334.1390419

[5] Michael Bendersky, Donald Metzler, and W Bruce Croft. 2011. Parameterized concept weighting in verbose queries. *Sigir* (2011), 605–614. https://doi.org/10.1145/2009916.2009998

[6] Michael Bendersky, Donald Metzler, and W Bruce Croft. 2012. Effective query formulation with multiple information sources. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 443–452.

[7] Abraham Bookstein and Don R Swanson. 1974. Probabilistic models for automatic indexing. *Journal of the Association for Information Science and Technology* 25, 5 (1974), 312–316.

[8] Mohamed Reda Bouadjenek, Hakim Hacid, and Mokrane Bouzeghoub. 2016. Social networks and information retrieval, how are they converging? A survey, a taxonomy and an analysis of social information retrieval approaches and platforms. *Information Systems* 56 (2016), 1–18. https://doi.org/10.1016/j.is.2015.07.008

[9] Oscar Chaparro and Andrian Marcus. 2016. On the Reduction of Verbose Queries in Text Retrieval Based Software Maintenance. (2016), 716–718. https://doi.org/10.1145/2889160.2892647

[10] Ronan Cummins. 2016. A Study of Retrieval Models for Long Documents and Queries in Information Retrieval. (2016).

[11] Emanuele Di Buccio, Massimo Melucci, and Federica Moro. 2014. Detecting verbose queries and improving information retrieval. *Information Processing and Management* 50, 2 (2014), 342–360. https://doi.org/10.1016/j.ipm.2013.09.003

[12] Dan Guo and Pengfei Gao. 2016. Complex-query web image search with concept-based relevance estimation. *World Wide Web* 19, 2 (2016), 247–264. https://doi.org/10.1007/s11280-015-0357-x

[13] Samuel Huston and W Bruce Croft. 2010. Evaluating verbose query processing techniques. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 291–298.

[14] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. *Kdd '02* (2002), 133–142. https://doi.org/10.1145/775047.775067

[15] Jae Dong Kim, Hema Raghavan, and Rukmini Iyer. 2010. Predicting Term Importance in Queries for Improved Query-Ad Relevance Prediction . *Learning* (2010).

[16] Marijn Koolen, Toine Bogers, and Jaap Kamps. 2016. Overview of the SBS 2015 suggestion track. *CEUR Workshop Proceedings* 1391 (2016).

[17] Giridhar Kumaran and James Allan. 2008. Effective and efficient user interaction for long queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 11–18.

[18] Giridhar Kumaran and Vitor R. Carvalho. 2009. Reducing long queries using query quality predictors. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09* (2009), 564. https://doi.org/10.1145/1571941.1572038

[19] Matthew Lease, James Allan, and W. Bruce Croft. 2009. Regression rank: Learning to meet the opportunity of descriptive queries. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5478 LNCS (2009), 90–101. https://doi.org/10.1007/978-3-642-00958-7_11

[20] Shih Hsiang Lin, Ea Ee Jan, and Berlin Chen. 2011. Handling verbose queries for spoken document retrieval. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2011), 5552–5555. https://doi.org/10.1109/ICASSP.2011.5947617

[21] Rachel Tsz-Wai Lo, Ben He, and Iadh Ounis. 2005. Automatically Building a Stopword List for an Information Retrieval System. *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)* 5 (2005), 17–24. http://eprints.gla.ac.uk/14094/

[22] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. 2005. Terrier information retrieval platform. In *European Conference on Information Retrieval*. Springer, 517–519.

[23] Jiaul H. Paik and Douglas W. Oard. 2014. A Fixed-Point Method for Weighting Terms in Verbose Informational Queries. *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14* (2014), 131–140. https://doi.org/10.1145/2661829.2661957

[24] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 275–281.

[25] Stephen E Robertson and K Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information science* 27, 3 (1976), 129–146.

[26] J.J Rocchio. 1965. Relevance feedback in information retrieval. *Prentice Hall* (1965), 313– 323.

[27] Gerard Salton, Edward a. Fox, and Harry Wu. 1983. Extended Boolean information retrieval. *Commun. ACM* 26, 11 (1983), 1022–1036. https://doi.org/10.1145/182.358466 arXiv:arXiv:1011.1669v3

[28] Mona Taghavi, Ahmed Patel, Nikita Schmidt, Christopher Wills, and Yiqi Tew. 2012. An analysis of web proxy logs with query distribution pattern approach for search engines. *Computer Standards and Interfaces* 34, 1 (2012), 162–170. https://doi.org/10.1016/j.csi.2011.07.001

[29] Xiaobing Xue and W Bruce Croft. 2012. Generating Reformulation Trees for Complex Queries. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2012), 525–534. https://doi.org/10.1145/2348283.2348355

[30] Xiaobing Xue, Samuel Huston, and W. Bruce Croft. 2010. Improving verbose queries using subset distribution. *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10* (2010), 1059. https://doi.org/10.1145/1871437.1871572

[31] Xueliang Yan, Guanglai Gao, Xiangdong Su, and Hongxi Wei. 2012. Hidden Markov Model for Term Weighting in Verbose Queries. (2012), 82–87.

[32] Bishan Yang, Nish Parikh, Gyanit Singh, and Neel Sundaresan. 2014. A study of query term deletion using large-scale E-commerce search logs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8416 LNCS (2014), 235–246. https://doi.org/10.1007/978-3-319-06028-6_20