Utility of Missing Concepts in Query-biased Summarization

Sheikh Muhammad Sarwar[1], Felipe Moraes[2], Jiepu Jiang[3], James Allan[1]
Center for Intelligent Information Retrieval
College of Information and Computer Sciences, University of Massachusetts Amherst[1]
Delft University of Technology[2]
University of Wisconsin-Madison[3]
smsarwar@cs.umass.edu,f.moraes@tudelft.nl,jiepu.jiang@wisc.edu,allan@cs.umass.edu

ABSTRACT

Query-biased Summarization (QBS) aims to produce a query-dependent summary of a retrieved document to reduce the human effort for inspecting the full-text content. Typical summarization approaches extract document snippets that overlap with the query and show them to searchers. Such QBS methods show relevant information in a document but do not inform searchers what is missing. Our study focuses on reducing user effort in finding relevant documents by exposing the information in the query that is missing in the retrieved results. We use a classical approach, DSPApprox, to find terms or phrases relevant to a query. Then, we identify which terms or phrases are missing in a document, present them in a search interface, and ask crowd workers to judge document relevance based on snippets and missing information. Experimental results show both benefits and limitations of our method compared with traditional ones that only show relevant snippets.

CCS CONCEPTS

• **Information systems → Summarization**; *Query intent*; Document topic models;

KEYWORDS

Query-biased summarization; Topic modeling

ACM Reference Format:
Sheikh Muhammad Sarwar[1], Felipe Moraes[2], Jiepu Jiang[3], James Allan[1]. 2021. Utility of Missing Concepts in Query-biased Summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463121

1 INTRODUCTION

Query-biased Summarization is a popular technique for presenting a document's content adaptively for a search query [1, 6, 8]. In a typical search system, a user views the summary of each search results first and then clicks on the result link to accesses the full content. A search result summary is helpful if a searcher can estimate the relevance of a document based on it. Previous studies showed that query-biased summaries [6], usually consist of a few sentences around query terms in the results, help searchers' click decisions better than static summaries.

However, search result snippets do not always help a user make the correct click decisions. We hypothesize that a user would be able to make better decisions if we design the summary more critically—instead of only showing the matched query terms in a document, we propose to also display the relevant concepts that are *missing* in the result. Here we define important query aspects as concepts. Commercial search engines such as Google shows missing query terms but not the missing concepts associated with a search query. There is no published study that provides models to generate *missing concepts* and evaluates their effectiveness for helping users' click decisions.

We design and evaluate a technique for generating and presenting missing concepts in QBS. Our approach uses a popular query topic modeling method DSPApprox [4] to learn query-related unigrams and phrases and then compare them to a document's content to identify missing concepts. We evaluate query-biased summaries, with and without showing the missing terms, using a crowd-sourcing user study. The rest of the article describes our method, experiment, and findings.

2 RELATED WORK

Search engines extract and present search result summaries to reduce user effort to find relevant documents [3]. Tombros and Sanderson [8] proposed to use Query-biased Summaries (QBS) alongside document title. Their approach was to extract document sentences with high coverage of query terms as summaries. Spirin and Karahalios [7] proposed an unsupervised extraction-based approach to generate structured snippets for a job search engine containing different facets of a job. Zhang et al. [10] applied a structured summarization approach for structured document search. These studies all found that QBS significantly improves both the effectiveness and efficiency of user relevance assessment based on summaries. While these approaches found different structures for presenting document summaries, they extract sentences that maximize the likelihood of the query terms. Consequently, these summaries become advertisements for the documents so that users click on them. To the best of our knowledge, there is no study on summaries that explains non-relevance, which is the focus of our study.

Recently, Maxwell et al. [6] studied user experience with textual search summaries of different lengths. They found that longer and more informative snippets are perceived more useful by the users

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463121

but did not help with click accuracy. Kim et al. [3] did a similar study focusing on mobile web search and came to a similar conclusion that longer summaries increase search time but do not improve click accuracy. These attempts motivated our research because they showed the limitation of relevant information in QBS—displaying more relevant information might not practically help users' click decisions even though users perceive longer summaries as more useful. In contrast, we study the impact of presenting missing relevant information to complement the traditional QBS methods that focus on extracting relevant snippets.

3 MISSING CONCEPT GENERATION

We make a simplifying assumption by considering query topics as concepts, and we apply a query topic modeling technique to extract missing concepts. For a search query, we first extract a list of topic terms – i.e., unigrams and phrases – related to the query's topic – from its top-retrieved documents. Then, we compare the topic terms with each document to identify the missing concepts.

We use DSPApprox [4] to obtain query-related topic terms. DSPApprox selects a small set of highly representative terms that best summarizes a set of documents. Dang and Croft [2] used this approach to find a hierarchical topic structure from top k ranked documents retrieved against a query. The algorithm constructs a vocabulary of unigrams and phrases from these documents. A sequence of words in a document becomes a phrase in the vocabulary if it matches any sequence of words in a Wikipedia entry's title. If an item in the vocabulary appears within a window of size w from a query term, the vocabulary item becomes a topic term. Each of these topic terms is scored based on its topicality and predictiveness. *Topicality* measures how informative a topic term is in terms of describing a set of documents. *Predictiveness* indicates how much the occurrence of a topic term predicts the occurrences of other terms. DSPApprox greedily selects a subset of topic terms maximizing the topicality and coverage of the vocabulary.

Once we find a set of topic terms $T = \{t_1, t_2, \ldots, t_n\}$ representing the topics of a query $q = \{q_1, q_2, \ldots, q_p\}$ of p keywords, we measure how each topic t_i relates to a document d_j using the following equation proposed by Dang and Croft [2]:

$$P(d_j \mid t_i) = P(t_i \mid d_j) \prod_{q_j \in q} P(q_j \mid d)^{\frac{1}{|t_i|+|q|}} \tag{1}$$

$P(d_j \mid t_i)$ indicates how prevalent a topic t_i is in a document d_j. Consequently, we can represent a document as a distribution over topics, $R(d_j) = [P(d_i|t_1), P(d_i|t_2), \ldots P(d_i|t_n)]$. We consider the k lowest scored topics from this distribution as the missing concepts and present those in the query-biased summaries.

4 EXPERIMENT

4.1 Experimental Design

We compare the effectiveness of Query-biased Summaries (QBS) with and without missing concepts in terms of assisting users using a crowd-sourced user study. In our study, a QBS of a document can consist of three components:

- The title (T) of the document
- A snippet (S) extracted from the document against the query
- Concepts Missing (M) from the document

We explore two QBS variants constructed from the components mentioned above. The first one is TS (Title + Snippet), which is provided by traditional web search systems. The second one, TSM (Title + Snippet + Missing Concepts), is our proposed variant that includes missing concepts. Our experiment seeks to answer the following research question: *What is the utility of providing missing concepts using the TSM variant*? Here we define the utility of missing concepts in a QBS as to what extent they help users more accurately assess a result's relevance without seeing its entire content.

To measure the utility of including missing concepts in QBS, we obtain user relevance judgments on two different variants of QBS. Given a query q, we retrieve ten documents for which we have relevance judgments. These relevance judgments are obtained from annotators who read the query topic, description, and narrative and judged the relevance R_{d_i} of a document d_i by reading the full content. We use R_{d_i} for the relevance judgment score for document d_i.

We follow the approach of Tombros and Sanderson [8] to evaluate the effectiveness of QBS. Given the summary of a document d_i retrieved against a query q, we ask a crowd worker to provide relevance judgment R'_{d_i} solely based on the summary and examine whether or not it is consistent with the judgment based on the full content. Now, if $R'_{d_i} = R_{d_i}$—i.e., the relevance judgments based on the summary and the whole contents are same, we consider that the summary was useful. We refer to R'_{d_i} as predicted relevance judgments. We consider binary relevance judgments with the two classes being relevant and non-relevant. We compute metrics such as accuracy and the confusion matrix based on the predicted and true relevant judgments.

To compute classification metrics, we need an equal number of relevant/non-relevant documents in a ranked list. Having a balanced ranked list means we can analyze workers' performances in both classes. To obtain such a balanced ranked list, we find rank k in the ranked list so that the next ten documents from that rank are uniformly distributed between relevant and non-relevant classes. Then we simply consider documents from rank 1 to $k - 1$ as non-existent in the corpus and compute our missing topic generation approach using documents starting from rank k. We do not make any changes to the ranking order.

4.2 Dataset and Model Parameters

We use Aquaint as the data collection in our experiment. Aquaint contains 1,033,461 news articles and has been used for the TREC 2005 Robust track [9]. The 2005 Robust track has focused on 50 poorly performing topics in an ad-hoc retrieval setting. For our study, we selected five topics used by Maxwell et al. [6]: 341 (Airport Security); 347 (Wildlife Extinction); 367 (Piracy), 408 (Tropical Storms); and 435 (Curbing Population Growth).

We indexed Aquaint with stopword removal and Krovetz stemming using `Indri` and removed near-duplicates and documents without a title. For near-duplicate detection, we used SimHash with parameters *blocks* = 4 and *distance* = 3, following the work of Manku et al. [5]. We also filtered the relevance judgments file accordingly, ignoring all documents that we removed in our pre-processing step. After this process, we ended up with 854,130 documents in our index.

We used `Indri` to generate the snippet component in our QBS. `Indri` generates snippets based on the best matching sentences with a query term a window of 50 words. The matched sentences are concatenated using ellipses. [1] To generate the missing concepts in our QBS, we found the best parameter setting for DSPApprox using manual inspection as there is no ground truth data for missing information generation. We extracted twenty topic terms for each query. The terms included both unigrams and multi-word phrases. We set minimum character and window size parameters of DSPApprox as 2 and 20, respectively.

We paid each crowd worker $2.50 USD to assess 10 results. To motivate quality judgments, we gave $1.00 USD bonus payment for those that achieved accuracy greater than 60%. We also removed workers having accuracy values ≤ 30%. After removal, we had in total 85 workers. For TS, we had 10, 10, 8, 4, and 9 workers for topics Airport Security, Wildlife Extinction, Piracy, Tropical Storms, and Curbing Population Growth, respectively. We had 9, 10, 10, 9, and 6 workers for TSM for the same sequence of topics. The topic *Tropical Storms* was particularly difficult to judge as very few workers could achieve above 30% accuracy.

4.3 Crowdsourcing Study Settings

We recruited workers from Amazon Mechanical Turk (MTurk). To recruit high-quality workers, we required our workers to have a HIT Approval Rate greater than 90%, be located in the USA, and have more than 1,000 approved HITs on Mturk. We had 48 and 50 workers for TS and TSM variants, respectively. We randomly assigned a worker to one of the variants and topics. We displayed a task description in Figure 1 and ten QBS to a worker. Figure 2 shows an example of a QBS with missing information generated from our system. We asked the workers to provide relevance scores on a scale from 0 to 5 based on QBS and later converted them to binary judgment using min-max normalization. We did this to compare it with the original binary relevance judgments from TREC 2005 relevance judgments.

Imagine you are a news reporter. Your editor has asked you to write a story on the following topic: **[search topic, e.g. *airport security*] : [search topic description] [search topic narrative]**. In order to write the story you will have to collect relevant documents about the topic. To facilitate this process we have provided you a ranked list of ten documents. However, we only provide a snippet or summary for each document rather than the whole content. Your task is to carefully read the summary of each document and then determine if the document would be relevant based on the information need of your editor. You will also have to specify at least three terms from the summary that motivated your decision.

Figure 1: Task template for our user study.

5 RESULTS

5.1 Missing Concept Utility Analysis

We set up our evaluation in such a way that we can apply standard binary classification metrics to analyze the utility of missing concepts in QBS. We evaluate TS and TSM using True Negatives (TN), False Positives (FP), False Negatives (FN), True Positives (TP), and Accuracy. We have true relevance judgments for those documents based on their full content and obtain crowd worker judgments based on QBS constructed using TS and TSM. Please refer to section 4.1 for a discussion on TS and TSM and the process to generate a ranked list with uniform distribution of relevant and non-relevant documents. We used a balanced dataset with the same number of relevant and non-relevant results, which is appropriate to evaluate TS and TSM using binary classification metrics.

Clinton Asks for More Funds to Fight Terrorism
...fight terrorism and to improve U.S. airline SECURITY in general. "Terrorists don't wait and neither should we," Clinton told reporters at a White House ceremony at which he received a report on AIRPORT SECURITY and urged Congress to act before it...checks on employees with access to secure AIRPORT areas. Endorsing all these measures, Clinton said he wanted Congress to provide money to go beyond the narrow issue of airline SECURITY and fight terrorism more broadly at home...
The document is missing the following potentially useful terms:
–metal detector –luggage –terminal –hijack –plane

Figure 2: Example QBS for topic Airport Security

Table 1 reports the results of our evaluation. One key point to notice here is that TS helps workers find relevant documents while TSM helps them filter out non-relevant documents. On average, TSM has fewer False Positives (FP) than TS, which is consistent with our expectations. For three of the queries (Query ID: 341, 347, 367), TSM has noticeably fewer FPs than TS, while for the other two queries (Query ID: 435, 408), TSM and TS have comparable FPs. It also shows that the presentation of missing concepts makes users conservative in judging a document as relevant. The average number of True Positives (TP) is comparatively higher for TS compared to TSM. The accuracy values are also higher in three cases among five for TSM.

Even though we can not conclude about the effectiveness of TSM based on the numbers reported in Table 1, we observe that workers become more careful about making relevance judgments with missing concepts. This phenomenon may be beneficial in application scenarios where user frustration increases by landing into a non-relevant document. For example, there are many seemingly relevant documents in the ranked list of a web search engine, and TSM may help users filter out the false positives. Our findings suggest that a large-scale study in such a setting with different missing concept generation approaches will be interesting.

5.2 Relation of Performance and Time

For both TS and TSM, we also recorded the time workers took to complete the annotation of a ranked list, i.e., ten document summaries against a search query. We computed the Pearson's Correlation Coefficient (PCC) between the time spent on a ranked list and the accuracy of the workers but did not find any significant correlation (R=-0.171 for TS and R=-0.051 for TSM). We noticed that, on average, workers took more time to judge TSM summaries than TS ones. They took 522 seconds on average to finish all the judgments for TSM compared to 395 seconds for TS. We believe workers took more time because TSM provided more information than TS, but their accuracy in judging was not related to time.

[1] Please refer to the Indri C++ API for more details: https://www.lemurproject.org/doxygen/lemur/html/classindri_1_1api_1_1SnippetBuilder.html

Query ID	Query	True Negatives		False Positives		False Negatives		True Positives		Accuracy	
		TS	TSM	TS	TSM	TS	TSM	TS	TSM	TS	TSM
341	Airport Security	3.6	**4.3**	1.4	**0.7**	**2.2**	2.7	**2.8**	2.3	0.64	**0.66**
347	Wildlife Extinction	3	**3.1**	2	**1.9**	**2.2**	2.5	**2.8**	2.5	**0.58**	0.56
367	piracy	4.22	**4.8**	0.78	**0.2**	**4.44**	4.6	**0.56**	0.4	0.48	**0.52**
435	curbing population growth	**3.11**	3.1	**1.89**	1.9	**2.33**	2.7	**2.67**	2.3	**0.58**	0.54
408	tropical storms	**1.5**	1.4	**3.5**	3.6	3.1	**2.9**	1.9	**2.1**	0.34	**0.35**

Table 1: Results for classification metrics for five queries

Query	Missing topics with (#relevant, #non-relevant) documents	S^q_{mi}	True Neg (TS/TSM)	Accuracy (TS/TSM)
Airport Security	terminal (3,4) | luggage (3,3) | metal detector (5,4) | plane (1,3) | hijack (1,4) | flight (1,3) | landing (2,2) | airline (2,1) | palestinian (2,0) | terrorist (2,0) | debt security (1,1) | passenger (2,0)	0.83	3.6/**4.3**	0.64/**0.66**
Wildlife Extinction	whale (4,5) | habitat (2,1) | endanger (2,2) | bird (5,2) | species (2,0) | wild (3,3) | natural (1,1) | tibetan culture (3,3) | tiger (1,1) | animal (2,2) | protect (0,2) | fish (0,3)	0.33	3/**3.1**	0**0.58**/0.56
Piracy	disc (3,5) | intellectual (4,2) | cds (4,4) | copyright piracy (1,1) | infringe (4,5) | software (4,2) | compact (2,2) | pirate (2,3) | music (0,1) | software piracy (1,0)	0.6	4.38/**4.8**	0.48/**0.52**
Curbing Population Growth	birth (4,3) | birth rate (4,3) | reproductive (4,4) | family plan (4,1) | increase (2,1) | development (1,1) | social (1,2) | world population (1,3) | percent (1,1) | country (1,1) | growth rate (1,0) | population growth rate (1,0) | billion (0,1) | reach (0,2) | people (0,1) | children (0,1)	0.44	**3.11**/3.1	**0.58**/0.54
Tropical Storms	northeast (4,2) | eastern (0,1) | flood (1,2) | late (0,1) | east (0,1) | weaken (5,3) | mile (1,1) | damage (2,0) | coast (1,2) | near (1,2) | hit (1,2) | evacuate (2,1) | island (1,0) | wind (1,0) | west (0,1) | hurricane center (2,1) | hurricane (2,1) | expect (0,2) | rain (1,1) | people (0,1)	0.4	**1.5**/1.4	0.34/**0.35**

Table 2: Missing topics shown to the workers for each query and their frequency in relevant and non-relevant documents.

For TSM, we asked the workers about the helpfulness of the negative information on a scale from 1-5. We observed negative information helpfulness has a weak ($R = 0.315$) but significant correlation ($p < 0.05$) with accuracy. For TS, we asked workers to select terms from the snippets that were helpful. We wanted to observe if there is an overlap between the missing information and terms or phrases selected from snippets. Because if a term in a snippet helps a worker decide on the relevance of a document, then using that term as missing information for any other document might be meaningful.

For a query q, we created an aggregated set of terms, T^q_{ts}, selected by the users from the TS version of QBS. We also have the set of missing concepts T^q_{tsm} computed against the query q. We compute a score, S^q_{mi} for the missing terms as $\frac{T^q_{ts} \cap T^q_{tsm}}{T^q_{tsm}}$. Table 2 reports the scores for each of the queries. We observed that TSM resulted in better accuracy and true negative values compared to TS when our missing concept generation technique was successfully finding terms that workers felt were important to decide on relevance. Specifically, for the queries *Airport Security* and *Piracy* we see a large gain in terms of both the metrics. Term annotation for this small set of queries can be useful for validating a missing concept generation approach or, in general, an approach that discovers topical terms related to a query.

6 CONCLUSION

We described a pilot study investigating the usefulness of showing missing concepts in QBS. We proposed and implemented a technique for extracting missing concepts based on DSPApprox, and we evaluated its effectiveness using a crowd-sourcing user study. Experimental results showed that missing concepts could be helpful to users' relevance judgments in some cases (queries) and across different evaluation metrics, but the overall benefits seem inconsistent. In contrast, our experiment also found that showing missing concepts can increase the effort of relevance judgments. To sum up, it requires further investigation to understand its usefulness and limitations fully. However, we contribute to the current understanding of search result summarization techniques by presenting the first results in this topic.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #IIS-2039449, and in part by NSF grant number 1813662. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

REFERENCES

[1] Wei-Fan Chen, Shahbaz Syed, Benno Stein, Matthias Hagen, and Martin Potthast. 2020. Abstractive Snippet Generation. In *Proceedings of The Web Conference 2020*.

[2] Van Dang and W. Bruce Croft. 2013. Term level search result diversification. In *The 36th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '13, Dublin, Ireland - July 28 - August 01, 2013*. 603–612.

[3] Jaewon Kim, Paul Thomas, Ramesh Sankaranarayana, Tom Gedeon, and Hwan-Jin Yoon. [n. d.]. What Snippet Size is Needed in Mobile Web Search?. In *CHIIR '17*.

[4] Dawn Lawrie, W. Bruce Croft, and Arnold Rosenberg. 2001. Finding Topic Words for Hierarchical Summarization. In *SIGIR '01*.

[5] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. 2007. Detecting near-duplicates for web crawling. In *WWW '07*.

[6] David Maxwell, Leif Azzopardi, and Yashar Moshfeghi. 2017. A Study of Snippet Length and Informativeness: Behaviour, Performance and User Experience. In *SIGIR '17*.

[7] Nikita Spirin and Karrie Karahalios. 2013. Unsupervised approach to generate informative structured snippets for job search engines. In *WWW '13*.

[8] Anastasios Tombros and Mark Sanderson. 1998. Advantages of Query Biased Summaries in Information Retrieval. In *SIGIR '98*.

[9] Ellen M Voorhees. 2006. The TREC 2005 robust track. In *ACM SIGIR Forum*. ACM New York, NY, USA.

[10] Lanbo Zhang, Yi Zhang, and Yunfei Chen. 2012. Summarizing highly structured documents for effective search interaction. In *SIGIR '12*.