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]
Virginia Institute of Technology[3]
{smsarwar,allan}@cs.umass.edu,f.moraes@tudelft.nl,jpjiang@vt.edu

ABSTRACT

Query Biased Summarization (QBS) aims to produce a summary of the documents retrieved against a query to reduce the human effort for inspecting the full-text content of a document [8]. Typical summarization approaches extract a document text snippet that has term overlap with the query and show that to a searcher. While snippets show relevant information in a document, to the best of our knowledge, there does not exist a summarization system that shows what relevant concepts is missing in a document. Our study focuses on the reduction of user effort in finding relevant documents by exposing them to omitted relevant information. To this end, 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 this approach.

KEYWORDS

Query Biased Summaries, Topic Modeling

ACM Reference Format:
Sheikh Muhammad Sarwar[1], Felipe Moraes[2], Jiepu Jiang[3], James Allan[1]. 2019. Utility of Missing Concepts in Query Biased Summarization. In *CIKM '20: ACM International Conference on Information and Knowledge Management, October 19–23, 2020, Ireland, Galway*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Query Biased Summarization is a popular technique for presenting a document's content adaptively with respect to a search query [1, 6, 8]. In a typical search system, a user views the summary of each of the search results at first and then accesses the full content of a document if they are intrigued by it's summary. Thus the utility of such a snippet is high if a searcher can assess the relevance of a document based on it. But, QBS generation methods locate the query terms within a document and extract their sentence contexts as summaries – biasing it in favor of the document [6].

SIGIR '21, 11-15 July, Online

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

As a consequence, even though snippets are useful in general, they do not always help a user to make the correct click decisions. We hypothesize that a user would be able to make better decisions if we design the summary in a more critical manner—instead of only showing the matched query terms in a document, we propose to also display the concepts that are *missing* in the document. In our work, important query aspects are defined as concepts. Commercial search engines such as Google shows missing query terms but not the missing concepts associated with a search query. To the best of our knowledge, there is no published study that characterizes *missing concepts*, provides models to generate them and studies their effectiveness.

We design and evaluate a technique for generating and presenting missing concepts in QBS. Our technique 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 a 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. All that work discovered 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 maximizes 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.

Recently, Maxwell et al. [6] conducted a study on the user experience with different length textual search summaries. They found that longer and informative snippets are perceived useful by the users but with no improvement in search accuracy. Kim et al. [3] did a similar study focusing on mobile web search and came to a similar conclusion longer summaries increase search time but do not improve search accuracy. These attempts motivated our research

because they showed that increasing information in one vertical is not going to help the users even if they perceive it as useful. We study the impact of presenting missing relevant information which is a different vertical.

3 MISSING CONCEPT GENERATION

We use a query topic modeling approach to extract missing concepts related to a query in its top-retrieved documents. For a search query, we first extract a list of representative unigrams and phrases related to the query's topic from its top-retrieved documents. Then, we compare the extracted unigrams and phrases with each document to identify the missing concepts.

We use DSPApprox [4] for extracting query-related topic representation. 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 a ranked list of documents retrieved against a query. The algorithm constructs a vocabulary of terms and phrases from these documents. If a sequence of terms matches another sequence of terms in a Wikipedia title, that sequence is considered a phrase and included in the vocabulary. 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, while predictiveness indicates how much the occurrence of a topic term predicts the occurrences of other terms. Finally, DSPApprox greedily selects a subset of topic terms maximizing the topicality and coverage of the vocabulary.

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

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

$P(d_j \mid t_i)$ 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 summary.

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 above mentioned components. 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*?

In order 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 whole 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 equal number of relevant/non-relevant documents in a ranked list. Having a balanced ranked list means we can analyze workers performances on both the 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 change 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 duplicate documents as well as 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 that have a query term in a window of 50 terms and those matching sentences are concatenated using ellipses. [1] To generate the

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

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 worker $2.50 USD and 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 $\leq$ 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 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 obtain a high quality pool of workers, we required our workers to have a HIT Approval Rate greater than 90%, be located in USA and have approval of more than 1,000 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 as shown in Figure 1 and ten QBS to a worker. Example of a QBS with missing information generated from our system is provided in Figure 2. We asked the workers to provide relevance score for a document 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 crowdworker judgments based on QBS constructed using TS and TSM. Please refer to section 4.1 for a discussion on TS and TSM as well as the process to

> **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

generate a ranked list with uniform distribution of relevant and non-relevant documents. As we have a balanced dataset based on binary judgments which justifies our choice of evaluating TS and TSM using binary classification metrics. Please note that we did not perform a within-subject study to obtain judgments meaning that QBS generated using TS and TSM for a specific query were inspected by different users.

The results of our evaluation is provided in Table 1. One key point to notice here is TS helps workers to find relevant documents while TSM helps them to filter out non-relevant documents. The average number of False Positives (FP) are generally lesser for TSM compared to TS, which is expected. For three of the queries (Query ID: 341, 347, 367) it is reasonably lesser and for two queries (Query ID: 435, 408) its pretty close to what TS achieves. It also shows that presentation of missing concepts make users conservative in judging a document as relevant. The average number of True Positives (TP) are 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 decision with missing concepts. This phenomenon is desired in the web search setting where user frustration increases by landing into a non-relevant document. There are many seemingly relevant documents in the ranked list of a web search engine and helping users to filter out the false positives is quite important. 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 measured the amount of 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 value (R=-0.171 for TS and R=-0.051 for TSM). We noticed that on an average workers took more time to annotate summaries generated by TSM rather than TS. They took 522 seconds on an average for TSM compared to 395 seconds on an average for TS. It shows that workers took more time because more information was available, but their accuracy in judging was not related to time.

For TSM, we asked the workers about the helpfulness of the negative information on a scale from 1-5. We observed negative

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.

information helpfulness has a weak and significant correlation with accuracy. The PCC value was 0.315 which is was significant at p<0.02. For TS, we asked workers to select terms from snippet 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 to make a decision about the relevance of a document, then using that term as missing information for any other document might be meaningful.

For a query q we create an aggregated set of terms, T^q_{ts}, selected by the users from TS based 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 $\frac{T^q_{ts} \cap T^q_{tsm}}{T^q_{tsm}}$. The scores for each of the queries are reported in Table 2. It can be observed that when our missing concept generation technique was successfully finding terms that workers felt were important to decide on relevance, TSM resulted in better accuracy and true negative values compared to TS. 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 can be helpful to users' relevance judgments in a number of cases (queries) and across a number of 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 fully understand its usefulness and limitations. 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.*