Summarization Search: A New Search Abstraction for Mobile Devices

Sunandan Chakraborty
New York University
sunandan@cs.nyu.edu

Zohaib Jabbar
New York University Abu Dhabi
zohaib.jabbar@nyu.edu

Lakshminarayanan Subramanian
New York University
lakshmi@cs.nyu.edu

1. INTRODUCTION

Mobile users in developing regions are known to have a relatively poor web experience [7] due to three fundamentally limiting factors: limited network bandwidth, high cost of connectivity and high latency. Several measurement studies have demonstrated web page load times in the order of several seconds to a few minutes [4] triggered due to poor network connectivity and high end-to-end latencies for even simple DNS lookups. Several prior efforts have proposed important optimizations at different network layers including the design of new transport protocols [10], new caching protocols [5] [7] [8] , middlebox strategies [11],delay tolerant network solutions [8] and rethinking the design of the applications to operate in such extreme environments [6] [9] [1] [7] [2].

In this paper, we propose *s*ummarization search as a new search abstraction for mobile users in developing regions built around the goal of minimizing the need for interaction and exploration involved in a standard web search task. The summarization search engine is implemented as an additional layer above a standard search engine model. Hence, it uses the state-of-the-art ranking, indexing and retrieval as offered by standard search engines. The contribution of this paper is not in these standard procedures of web search but presenting the results in a novel way which potentially can improve the experience of searching the web from mobile devices. This abstraction is specifically designed for *focused* search queries, where the search action of the user relates to a specific information need. The goal is to provide a user with her information needs within a single round of interaction. Given a query, our summarization search service interacts with conventional search services (like Google, Bing etc.), analyzes the contents of the top search result pages and provides a condensed and summarized search response to the user, highlighting the essential parts – with respect to the query – from the target documents. In essence, the users can find their appropriate search result in at most one or two clicks, one for the search and one for choosing the result page.

ACM DEV 2015, December 1–2, 2015, London, United Kingdom.
ACM 978-1-4503-3490-7/15/12.
DOI: http://dx.doi.org/10.1145/2830629.2835217.

We evaluated the summarization search interface using a set of 400 queries from users using Mechanical Turk crowdsourcing platform and asked the users to rate the summarized search results. The users were asked to rate the results in a scale of 5 (1 is high and 5 is low), where around 85% of the queries received an average rating of 1 or 2. The user study also showed that in 55% of the cases the users found the information in the summarized search result page, without the need for further browsing. This demonstrates the effectiveness of the summarization search, where in more than half the cases users can obtain the information with just one operation of submitting the query. Although many of the commercial search engines can do the same for very specific types of queries, our system is capable of performing this for generic web search queries.

2. METHODOLOGY

The summarization engine uses the Google Search API to download the the top 64 (configurable) search result pages. The key goal of the summarization engine is to perform detailed text analysis of all the result pages to prepare a condensed summary page across all the search result pages. Text Summarization is a process of creating a concise version of a larger text preserving the main theme of the original text. There are different ways of summarizing a text. Two popular methods used in the NLP literature are Extraction based summarization and abstraction based summarization [3]. Our text summarization uses the extraction based summary approach which involves creating the summary by using exact sentences from the original text. Unlike the standard task the extraction in our case is driven by the query terms, highlighting the portion of the documents which have high relevance with respect to the query terms. Our extraction based summarization algorithm has two steps: (a) identification of key terms in a document w.r.t. the query terms; (b) identification of key portions of a documents using the key terms identified in the previous step.

To identify important key terms in a document we use a natural language parser to parse each sentence. This step breaks down each sentence into noun phrases, verb phrases and prepositional phrases etc. The noun phrases are the most informative parts of the sentences and the verbs – which depict the actions – have relevance too. A popular measure to determine importance of a term is the term frequency-inverted document frequency (TF-IDF). We computed IDF of the noun and verb phrases as the web probability of them using the Microsoft Ngram Web Service [1]. Let $P_{web}(t_i)$ represent this measure for the phrase t_i. High value of the web probability represents the popularity of the phrase. This means it has relatively less importance in this context because it tends to appear very frequently in any large English corpus of text. $P_{corpus}(t_i)$ is the normalized frequency of t_i. This score is similar to TF-IDF but in this case all the documents is a focus set of documents resulting from a query. Hence, many important terms might have high document frequency and using TF-IDF weights will lower their scores. The algorithm for summarizing the search results is presented below.

Algorithm 1 Summarization Search

procedure SUMMARIZE(query = q, document= d)
  for each $t_i \in V$ **do**
    compute $Imp(t_i) = P_{corpus}(t_i)/P_{web}(t_i)$
    compute $Imp_{norm}(t_i) = \frac{Imp(t_i)}{\sum_i Imp(t_i)}$
  end for
  for each $s_j \in d$ **do**
    compute $Score(s_j) = \sum_{t_i \in s_j, V} w_t \times Imp_{norm}(t_i) + \sum_{t_q \in s_j, Q} w_q \times Imp_{norm}(t_q)$
  end for
  $\mathcal{S} \leftarrow sort([\forall s_j \in d], key = Score(s_j), order = Desc)$
  $summ\_score \leftarrow 0$
  $summary \leftarrow empty\_list$
  for $s_j \in \mathcal{S}$ **do**
    if $\Delta summ\_score_{k-1:k} \leq \epsilon$ **then**
      break
    else
      $summ\_score \leftarrow summ\_score + Score(s_j)$
      $summary.append(s_j)$
    end if
  end for
  $summary \leftarrow sort([\forall s_j \in summary], key = j, order = Asc)$
    $\triangleright$ j = sequence no. of the sentence in document d
  return summary
end procedure

3. EVALUATION

We performed a user study to both generate queries and evaluate the summarization search interface for those queries. In the first round of the user study, users were asked to submit queries which they are more likely to submit from mobile devices instead of desktop/laptop computers. In the second phase we engaged 30 users to evaluate the results from these queries. The goal was to evaluate the quality of the summarized search results. The user study was done using the Amazon Mechanical Turk platform and the chosen users were all Mechanical Turk Masters. Masters are workers who have demonstrated high accuracies in previous tasks and this ensured better quality in the user responses.

We asked the users to judge the summarized search responses based on the given query and rate the quality using a score between 1(highest) and 5(lowest). Every query was rated by all the 30 users. We computed the mean and mode of the scores of each query rated by all the users. Figure 1 shows the histograms of frequency distributions of mean and mode scores for all the queries. Around 45% of the queries received an average rating between 1 and 2 and more than 80% of the queries received positive ratings from the users.

[1]research.microsoft.com/en-us/collaboration/focus/cs/web-ngram.aspx

Figure 1: Histograms of ratings per query for all users – mean (left) and mode (right) rating per query

Table 1: Average ratings per query per user for different types of queries

Type of query	Average rating
Rating-type	2.25
Information-type	1.72
Location-type	2.32

Figure 1(right) shows that more than half (52%) of the queries received a score of 1 from majority of the users (i.e. mode score of 1). We also evaluated the performance for different types of queries in this study. We manually classified the queries into 3 sets – location-specific, ratings-type and information-type. Location specific queries are those where users asked about points of interests in (or *near*) a location (e.g. find the nearest movie theater in *location* etc.) Examples of rating type query is 'top restaurants". Finally, information type queries are those where users looked for specific information (e.g. "fastest land vehicle"). Table 1 summarizes the relative performance across these categories. The results show that the performance was best for information-type queries and worst for location-types. A probable explanation is that information-type had very specific answers and we did not use any location services which might have resulted in some erroneous results.

4. REFERENCES

[1] Chen, J. and Subramanian, L. and Li, J. RuralCafe: web search in the rural developing world. WWW. 2009
[2] Teevan, J. and Ramage, D. and Morris, M. Ringel. # TwitterSearch: a comparison of microblog search and web search. WSDM. 2011
[3] Radev, D. R., Hovy, E., and McKeown, K. (2002). Introduction to the special issue on summarization. Computational Linguistics., 28(4)
[4] Y. Zaki, J. Chen, T. Pötsch, T. Ahmad, and L. Subramanian. Dissecting web latency in ghana. In *IMC 2014*.
[5] J. Chen and L. Subramanian. Interactive web caching for slow or intermittent networks. In *DEV 2013*.
[6] D. Iland and E. Belding. Emergenet: robust, rapidly deployable cellular networks. *Communications Magazine, IEEE*, 52(12), 2014.
[7] A. Balasubramanian, N. Balasubramanian, S. J. Huston, D. Metzler, D. J. Wetherall. Findall: A local search engine for mobile phones. *CoNEXT 2012*.
[8] S. Isaacman and M. Martonosi. The c-link system for collaborative web usage: A real-world deployment in rural nicaragua. In *NSDR 2009*.
[9] W. Thies, J. Prevost, T. Mahtab, G. T. Cuevas, S. Shakhshir, A. Artola, B.D. Vo, Y. Litvak, S. Chan, S. Henderson, M. Halsey, L. Levison, and S. Amarasinghe Searching the World Wide Web in Low-Connectivity Communities. In *WWW*, 2002.
[10] Y. Zaki, T. Potsch, J. Chen, L. Subramanian, C. Gorg Adaptive Congestion Control for Unpredictable Cellular Networks. *SIGCOMM 2015*.
[11] A. Balasubramanian, B. Levine, A. Venkataramani. DTN Routing as a Resource Allocation Problem. ACM Sigcomm, August 2007