Searching Large Textual Dataset With Limited Computational Resources

Anagha Kulkarni
San Francisco State University
ak@sfsu.edu

ABSTRACT
In this paper we propose a search approach that can process large volumes of textual data efficiently and effectively even in environments where computational resources are limited. The traditional search solution for large collections assumes availability of practically unlimited computational resources. For many applications and organizations this assumption is not realistic. Empirical evaluation of the proposed approach using some of the largest available datasets demonstrates that the proposed search approach is substantially more efficient than the existing approach, is on par if not better in terms of effectiveness, and can operate using very few computational resources.

AUDIENCE: [Information Retrieval/Large-scale Search] [Text Processing] [Advanced technical talk]

1. INTRODUCTION
The ability to search large volumes of textual data (billions of documents) for relevant information, and doing so efficiently and effectively, has been critically important for commercial search engine, such as, Google and Bing. However, this ability has become increasingly important for other applications as well. For instance, a central component of a clinical decision support system designed to assist medical professionals is a search engine that can wade through large volumes of medical records and medical literature searching for information that is relevant to the case at hand. More data typically generates a stronger candidate pool of answers or results, and provides more accurate selection of final results. In spite of abundant motivation, many applications have to either settle for working with only a subset of the available data (which degrades effectiveness), or accept slow search response, due to lack of necessary computing power. The goal of our presented work is to provide a search solution that can process large volumes of text data, efficiently and effectively, in resource-constrained environments.

The prevalent search solution for large datasets assumes availability of ample computing resources, where the dataset is partitioned into multiple smaller subsets (shards), each of which is assigned dedicated computing resources. At query execution time all shards are searched in parallel to provide short query response time [Barroso et al., 2003; Chowdhury and Pass, 2003, Cacheda et al. 2007, Moffat et al. 2007, Broccolo et al. 2013]. We refer to this approach as exhaustive search because the complete dataset (all shards) are searched for potential query results.

Our work presents an alternative approach, selective search, which partitions the dataset into shards in such a way that at query execution time only a few selected shards need to be searched for a query, thus reducing the computational requirements [Kulkarni and Callan 2010, 2014].

![Figure 1: Schematic Diagram of Selective Search](image)

2. SELECTIVE SEARCH
The selective search approach consists of two main phases. The first one relates to dataset partitioning (shard creation), which is depicted in the upper half of Figure 1. This is an offline process, that is, it is performed before any query is processed by the system. The second phase of selective search commences when the system receives a query. The shards that are likely to contain relevant documents for the query are identified and searched. The retrieval results obtained at each of the selected shards are merged to compile the final result set which is presented to the user. We describe these two phases next.

2.1 Shard Creation: Our shard creation approach is based on cluster hypothesis, as per which if similar documents are grouped together then the documents that are relevant to a query also get clustered together [Jardine and van Rijsbergen, 1971]. Partitioning the large dataset based on the similarity of the documents creates shards that are semantically homogenous, and each shard can be seen as representing a distinct topic (e.g. politics, sports, and finance). We refer to these partitions as topic-based shards. Based on cluster hypothesis we premise that topic-based shards concentrate relevant documents for a query into a few shards. For example, if the query is barack obama it is likely that the relevant documents for this query are in the politics shard, and not in the sports or finance shards.
To organize the documents based on their similarity we use a variant of the time-tested K-means clustering algorithm [Lloyd 1992]. At a high-level, the K-means algorithm starts by randomly selecting K documents as the seed centroids (centers) of K clusters. After this the algorithm alternates between the next two steps until some stopping criterion has reached. 1) A similarity metric is then used to identify the most similar centroid for every document in the dataset, and the document is assigned to the corresponding cluster. 2) Once all the documents have been assigned to the K clusters, each centroid is recomputed based on the documents assigned to the cluster. Although the computational complexity of the K-means algorithm is linear in the number of documents, applying this algorithm to very large collections can still be prohibitively expensive in resource-constrained environments because of the iterative nature of the algorithm. To solve this scalability problem we develop a sample-based K-means approach that applies the standard K-means algorithm to only a small subset of documents. The resulting K clusters and their centroids are then used as the definitions of the K topic-based shards. The remaining documents are assigned to clusters by finding the most similar cluster definition for each.

We also experiment with two other partitioning techniques that have been used in prior work: random [Puppin et al. 2010], and source-based [Voorhees 1995, Callan et al. 1999]. If the collection has to be partitioned into K shards, then the random allocation policy assigns each document to one of the K shards at random with equal probability. This is the most commonly employed sharding technique due to its simplicity, and scalability. However, on the downside, random partitioning spreads the relevant documents for a query across many shards, instead of concentrating them into a few shards. The source-based sharding approach organizes the collection into shards using the source of the document, such as, the company department that generated the document, or the Web host. To perform source-based partitioning we first sort the document URLs, which arranges documents from the same website consecutively, and then groups of \(|M/K|\) consecutive documents are assigned to each shard where \(|M|\) is the total number of documents in the collection, and \(K\) is the total number of shards to be created.

2.2 Shard Ranking: Once the dataset is partitioned into shards, the search system can start processing queries. However, for each query the shard(s) that contains majority of the relevant documents need to be first identified. For this task we employ a widely-used algorithm, ReDDE [Si and Callan 2001], which maintains a profile of the contents of each shard in order to estimate a relevance score of each shard for the query. The query is then processed at \(T\) shards with highest relevance scores. The search results obtained at each of the T shards are then merged, and the consolidated result list is presented to the user. This concludes the selective search process for a query.

3. EXPERIMENTAL DETAILS

Datasets: We used two of the largest evaluation datasets available to empirically evaluate the selective search approach. Specifically, we use the GOV2 dataset (25 million webpages from US government domains. Size: 0.4TB), and the ClueWeb09-English (CW09-Eng) dataset (500 million webpages crawled in early 2009. Size: 15TB). For GOV2 dataset, a set of 150 queries are available for which the dataset documents were manually annotated for relevance by NIST. For CW09-Eng a set of 200 queries with relevance judgement is available. The average query length for GOV2 queryset is 3.1 terms, and for CW09-Eng it is 2.1 terms. On an average there are 181, and 107 relevant documents per query for GOV2 and CW09-Eng, respectively.

Setup: For selective search experiments each dataset was partitioned into \(K\) shards using each of the three shard creation techniques described in Section 2.1 (Random: Rnd, Source-based: Src, Topic-based: Top). The value for \(K\) was chosen such that the average shard size would be half a million documents. Thus GOV2 was partitioned into 50 shards, and CW09-Eng into 1000.

Evaluation metrics: The search effectiveness is quantified using standard retrieval metrics: Precision@10 (P@10), P@30, and P@100. The P@X metric is the ratio of number of top X retrieved documents that are relevant and X. The search efficiency is quantified using a cost metric that measures the average number of documents evaluated per query.

4. RESULTS AND ANALYSIS

The results for the experiments that compare exhaustive search (baseline) with selective search are given in Tables 1 & 2 for GOV2, and CW09-Eng datasets, respectively. Some of the prominent trends that span across the two datasets are as follows. The partitioning approach strongly influences the performance of the selective search approach. The retrieval effectiveness of selective search with random shards is substantially lower than that of exhaustive search, especially so at deeper ranks (P@100).

<table>
<thead>
<tr>
<th>Search Method</th>
<th>T</th>
<th>Search Effectiveness</th>
<th>Search Cost (million)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@10</td>
<td>P@30</td>
<td>P@100</td>
</tr>
<tr>
<td>Exh All</td>
<td>0.53</td>
<td>0.48</td>
<td>0.38</td>
</tr>
<tr>
<td>Sel-Rnd 1</td>
<td>0.21</td>
<td>0.10</td>
<td>0.03</td>
</tr>
<tr>
<td>Sel-Src 1</td>
<td>0.34</td>
<td>0.25</td>
<td>0.15</td>
</tr>
<tr>
<td>Sel-Top 1</td>
<td>0.39</td>
<td>0.32</td>
<td>0.23</td>
</tr>
<tr>
<td>Sel-Rnd 5</td>
<td>0.38</td>
<td>0.27</td>
<td>0.13</td>
</tr>
<tr>
<td>Sel-Src 5</td>
<td>0.48</td>
<td>0.41</td>
<td>0.29</td>
</tr>
<tr>
<td>Sel-Top 5</td>
<td>0.53</td>
<td>0.47</td>
<td>0.36</td>
</tr>
<tr>
<td>Sel-Rnd 10</td>
<td>0.44</td>
<td>0.33</td>
<td>0.19</td>
</tr>
<tr>
<td>Sel-Src 10</td>
<td>0.52</td>
<td>0.45</td>
<td>0.33</td>
</tr>
<tr>
<td>Sel-Top 10</td>
<td>0.53</td>
<td>0.48</td>
<td>0.37</td>
</tr>
</tbody>
</table>

Table 1. Dataset: GOV2. K = 50, T: Number of top shards searched for each query. Boldface numbers are not statistically
Selective search with source-based shards demonstrates better ability to compete with exhaustive search, however, even this setup struggles to perform well at deeper ranks, especially for the larger dataset. The topic-based shards support the most competitive selective search performance. Although, costlier than random and source-based policies, the topic-based policies converge to exhaustive search effectiveness fastest. The best balance between search effectiveness and efficiency is provided by selective search with topical shards. When studying selective search's performance for individual evaluation metrics we see that very few top shards need to be searched (5 for GOV2, and 1 for CW09-Eng dataset) to be competitive at early ranks, as is illustrated by the trends for P@10 and P@30 metrics. To optimize the performance at deeper ranks (P@100), selective search has to search more shards for the smaller dataset (10 for GOV2), but for the larger dataset searching just the top shard is sufficient.

As compared to the search cost of exhaustive search, the selective search costs are substantially lower for all the selective search configurations and both datasets. For GOV2 the largest improvement in search efficiency is 71% at T=5 for P@10 metric, and the smallest improvement is 52% at T=10 for P@100. The improvements are much bigger for the larger datasets. 99% at T=1 for all three effectiveness metrics.

For CW09-Eng selective search is more effective than exhaustive search for P@10 and P@30, and these improvements at T=10 are statistically significant. This result suggests that topical partitioning of the collection is able to reduce noise (false positive/relevant documents) from the search space, which might be especially helpful for a noisy and heterogeneous datasets like CW09-Eng.

For both the datasets, as few as one computational core, or a handful cores will be sufficient to operationalize the search of such large datasets. Although exhaustive search can also be performed on such configurations the query response time would be much slower because substantially more computations are conducted by the exhaustive search approach for each query.

5. OUTCOMES/CONCLUSION

Overall, the selective search approach with topic-based shards provides a cost-effective alternative to the exhaustive search approach. Even the smallest reduction in search cost offered by selective search is substantial. As such, selective search can support efficient and effective search of large datasets in low-resource environments.

PARTICIPATION STATEMENT

I promise to attend GHC 2015.

BIO

The author is an assistant professor at the computer science department in San Francisco State University. She obtained her doctorate degree in 2013 from the Language Technologies Institute at Carnegie Mellon University. The work described here is based on her PhD dissertation research, a body of work that she is very well familiar with, and has presented to a wide range of audience.

REFERENCES/BIBLIOGRAPHY


<table>
<thead>
<tr>
<th>Search Method</th>
<th>T</th>
<th>Search Effectiveness</th>
<th>Search Cost (million)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P@10</td>
<td>P@30</td>
</tr>
<tr>
<td>Exh</td>
<td>All</td>
<td>0.10</td>
<td>0.11</td>
</tr>
<tr>
<td>Sel-Rnd</td>
<td>1</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>Sel-Src</td>
<td>1</td>
<td>0.03</td>
<td>0.02</td>
</tr>
<tr>
<td>Sel-Top</td>
<td>1</td>
<td>0.13</td>
<td>0.12</td>
</tr>
<tr>
<td>Sel-Rnd</td>
<td>5</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>Sel-Src</td>
<td>5</td>
<td>0.07</td>
<td>0.04</td>
</tr>
<tr>
<td>Sel-Top</td>
<td>5</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>Sel-Rnd</td>
<td>10</td>
<td>0.06</td>
<td>0.03</td>
</tr>
<tr>
<td>Sel-Src</td>
<td>10</td>
<td>0.09</td>
<td>0.05</td>
</tr>
<tr>
<td>Sel-Top</td>
<td>10</td>
<td>^0.12</td>
<td>^0.12</td>
</tr>
</tbody>
</table>

Table 2. Dataset: CW09-Eng. K=1000 shards. T: Number of top shards searched for each query. Boldfaced numbers: same as Table 1. *indicates statistically significant improvement over corresponding exhaustive search number.

The author has presented to a wide range of audience.
Subject: Congratulations! Your submission was accepted to GHC2015...
From: GHC 2015 Submissions <do_not_reply@linklings.com>
Date: 5/22/15, 6:33 PM
To: <ak@sfsu.edu>

Congratulations Anagha Kulkarni,

We are happy to announce that your submission, "Searching Large Textual Dataset With Limited Computational Resources", has been accepted to the 2015 Grace Hopper Conference. This email will outline your next steps.

Submission Title: Searching Large Textual Dataset With Limited Computational Resources
Session Title: Artificial Intelligence: Lightning Talks
Session Track: Artificial Intelligence
Session Day: Fri Oct 16

GHC General Sessions will run on Wednesday, October 14 from 10:30am – 3:15pm, on Thursday, October 15 from 10:30am – 5pm, and on Friday, October 16 from 10:30am – 3:30pm. Your session time will fall within these ranges. More info on your session’s exact time to come shortly.

***NEXT STEPS***

Sunday, June 28th, 2015 @ 11:59 PM (Pacific Standard Time): Deadline to “Confirm” your participation in the conference and to submit all the Conference Ready Material. If we do not receive confirmation of your participation and the Conference Ready Materials by this date, then you will forfeit your spot at the conference. No substitutions allowed.

More information available here -> http://gracehopper.org/information-for-speakers/

TELL EVERYONE YOU ARE SPEAKING AT GHC!

For additional questions and future communication, please contact Stuti Badoni, GHC Program Manager at stutib@anitaborg.org

Registration and hotel information will be available on the GHC website (www.gracehopper.org) around June 2nd.

Congratulations once again and we look forward to seeing you in Houston!
GHC 2015 Conference Leadership