Project Report

SAMPLE SIZES FOR QUERY PROBING IN DISTRIBUTED INFORMATION RETREIVAL

December 7, 2014

Rahul Ramakrishna
University of Massachusetts Amherst
Dept Of Computer Science
rahulram@cs.umass.edu

1 Introduction

In Distributed Information Reterival systems, information is held in separate collections, which might be in different physical locations or on separate servers. The query is first passed to a central broker. The broker then sends this query to all or some of the servers. The cost of networking during query execution seems an overhead, escepially if the query is sent to servers which dont have similar collections.

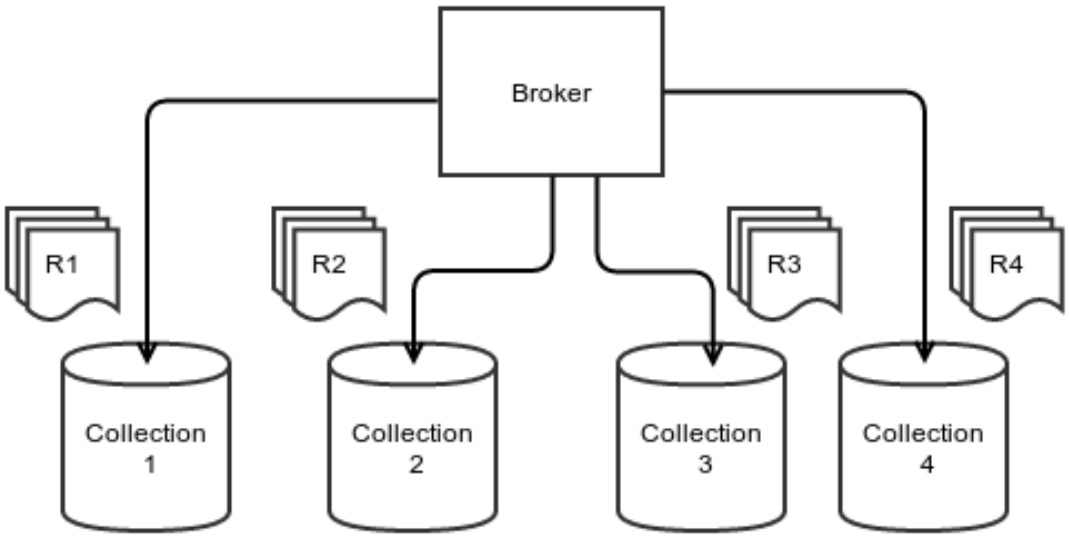

Figure 1: DIR System.

Thus, DIR has 2 major issues to be resolved.

1. Selection of a particular collection
2. Merging of Results.

In this report, we are mainly focusing on selection of collections. We are focusing on how to characterize a particular collection such that, the broker can decide on which servers to send the queries to further probe for results. We further explore 2 main sampling techniques which are used to represent collection of sets.

2 Query Probing

In non-cooperative environments like distributed systems. We dont get index information from each of the collections that are being fetched. Instead, its the brokers that constantly probe into the collection by sending artificial queries in random order and evaluate returned answers. The answers are called `probes` and the process is called `query`

`probing`. For example, if we send a series of single probe queries such as "books", "football", "ibm" and the following number of answers are returned: 1000, 200, 450 . From the results we can guess that, the documents are more likely to talk about Books and IBM rather than football.

The following is the algorithm initially proposed by Callan et al [1], which iteratively discovers the language model of the collections in non-cooperative environments.The language model is updated according to the new terms found in the retrieved documents. The next probe queries are selected from the obtained language model. Probing continues until a stopping criterion is met.

Algorithm 1 Query Sampling

```
1:  procedure QS
2:      Select an Initial Query Term
3:      Run Query on the IR System
4:      Reterive Top N Documents as Result.
5:      Update the Resource Description based on characterstics of retreived documents.
6:  loop:
7:      for words,freq in Documents
8:      Update Learned Resource Descriptions.
9:      if StopCriteria() = Yes then
10:         break;
11:     else
12:         Goto 3
```

Callan et al suggested that assigning $N = 4$ and $StopCriteria = 75$ Docs i.e examining a total of 300 documents will be a good representation of the collection. The algorithm has many open ended choices which needs to be tuned. For example, how to select query terms and how to select documents to examin per query and most importantly *when to stop sampling*. In the subsequent sections we will evaluate sampling techniques mentioned by Callan et al and adaptive query probing techniques by Milad et al [2].

3 Sampling Techniques

3.1 Static Sampling using Ctf

In static sampling technique. The paper uses a Character term frequency as a metric in order to calibrate the sample size. *Ctf* ratio is defined as the ratio of term occurences in the collection that are covered by terms in the learned resource description. For the learned vocublary V' and actual vocabulary V,

$$Ctf = \frac{\sum_{i \epsilon V'} ctf_i}{\sum_{i \epsilon V} ctf_i}$$

ctf_i is the number of times the term i occurs in the database (collection term frequency or ctf). The ctf ratio is computed after the stopwords were removed. Experiments

performed by Callan et al suggested that the *Ctf* curve usually smoothens at sample of 300 documents. Thus, making a general stopping criteria in query probing algorithms.

3.2 Adaptive Query Probing

In adaptive query technique, we extract the most significant terms from each collection by gathering all terms. Cosine $tf.idf \geq \gamma$ Where (γ) is a threshold value using Indrie. This information is used after termination of query-based sampling, as a measure of the effectiveness of the collected summaries and of the risk of missing significant terms.

$$Term = \begin{cases} Significant & \text{if } tf.idf \geq \gamma \\ Not\ Significant & \text{otherwise} \end{cases}$$

Queries from the Robust.qrels were choosen randomly. mostly, queries of size 2 - 4 terms were choosen, which were similar to web queries Jansen et al [3] . In this technique we make an assumption that collections only return a limited number of documents for any given query.

The Recall metric to measure completeness of a term set is defined as,

$$Recall(s, \gamma) = \frac{\text{No of Significant terms in sample}}{\text{Total No of Significant terms}}$$

we gather samples of different sizes ranging from 100 to 4000 documents to iteratively probe into them. In the further sections its explained that the the representation size of 4000 documents would meet the stopping criteria.

4 Measure of Effectiveness

All the experiments were performed on Robust dataset. We intend to verify the following hypothesis

1. As long as we keep sampling, the vocabulary continues to grow.
2. The rate of vocabulary growth is not a good way to estimate collection size.
3. The risk of missing significant terms is high with traditional sampling.

4.1 Ctf Ratios

Using the static sampling algorithm proposed by Callan et all. The Ctf metric is been calculated for the robust dataset. The fig 2 shows the ctf ratio for the sample sizes ranging from 100 to 2500 sampled documents.

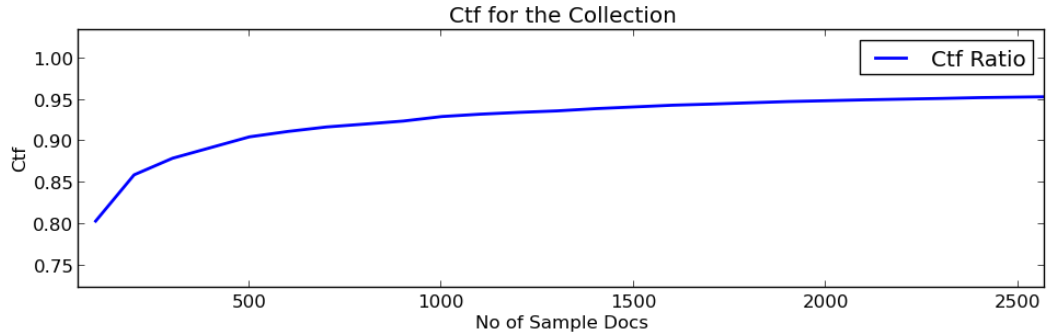

Figure 2: Ctf Ratio of Robust Collection.

The expected value of curve smoothing should be from 300. But as we can see from the graph, the smoothening of the curve begins from 500 documents. Thus, if probing is halted after sampling 300 documents, the risk of losing significant terms is high. However, this value could be different for different collections, which makes it even more difficult validate *Ctf* as a meteric to sample documents.

4.2 Significant terms

For calibrating siginificant terms, samples ranging from 100 to 4000 we extracted. Each sample n contains all the documents from sample n - 1, plus 100 new documents. The initial sample always extracts 100 distinct documents. At any given point, the system calculates the number of unique and significant terms available in the samples. We show results for 4000 documents.

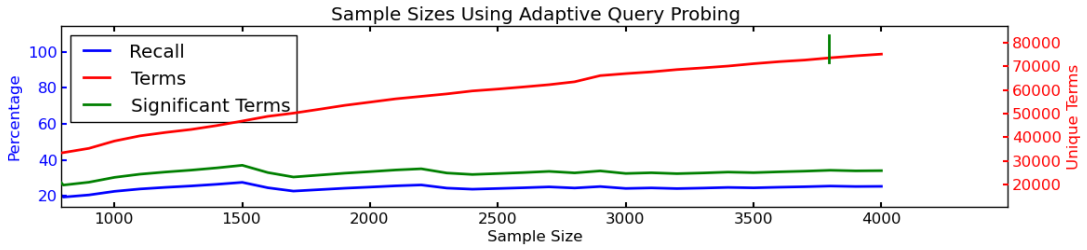

Figure 3: Statistics of Robust Collection.

From the fig 3 we can see that a sample of 4000 documents would suffice for experiment testbed. Since, after 4000, even though the vocabulary keeps growing, the addition of significant terms smoothens out. The green vertical line represents Convergence point, where slope of the lines are lesser than a certain threshold τ. For η subsequent samples the rate of growth in vocabulary becomes less than threshold τ . Query sampling reaches a good coverage of the collection vocabulary when the slope becomes $\leq \tau$.

4.3 Result Evaluation

We have created 3 sample sets for both the static and adaptive sampling techniques. At each iteration, we used a different set of random queries.

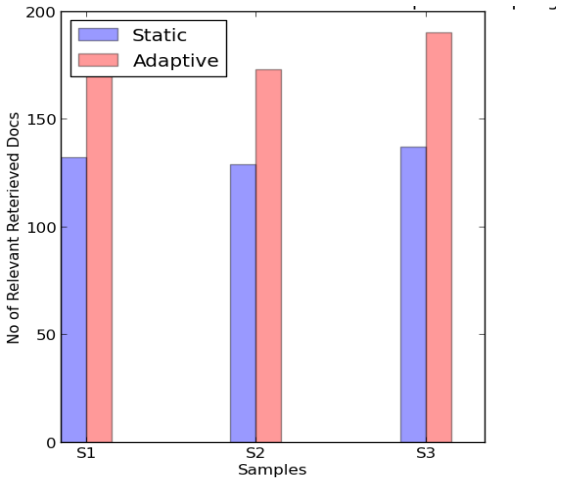

Figure 4: Relevant Retreived Docs for Static and Adaptive Sampling.

We use trec eval tool to evaluate the results produced by sample sizes. In the fig 4 , we compared 3 different sample sizes generated from static and adaptivefor different random set of queries. We observe that in general the number of retreived documents is higher for adaptive samples as apposed to static. The following table records MAP, P@10, P@20

Table 1: Evaluation Metric for Sample Sizes

Static Samples	MAP	P@10	P@20
S1	0.0466	0.1760	0.0880
S2	0.0394	0.1751	0.0921
S2	0.0576	0.2098	0.1170
Adaptive Samples			
S1	**0.0541****	0.2297	0.1149
S2	0.0411	0.2011	0.1103
S2	**0.0801****	**0.3813****	**0.2010****

As we can see, in general adaptive query produces better results that static samples. In some cases, they are significantly better than static samples. In the current adaptive query technique we use $\eta = 3$ and threshold $\tau = 2\%$, which means the probing was stopped after 3 consequetive probes having growth rate less than 2%.

5 Future Work

The adaptive query probing technique seems effective for characterizing collections. We could further tune the parameters η (no of consequetive probes) and τ (threshold) and evaluate the results. Decreasing η or increasing τ will probably lead to less effective representations of collections. Also, setting τ to a very low value will probably only lead to computationally expensive operations and may not give us better results than a higher τ value.

6 Conclusion

Adaptive Query probing is a better way for summarizing collections in Distributed Information Retreival. Static sampling sizes using ctf ratio leads to considerable loss of effectiveness. Adaptive query has a better strategy of knowing, when to stop query probing. The results indicate that, rate of arrival of new terms becomes constant and relatively few new significant terms of high impact in retrieval are observed. In the experiments conducted, the adaptive query probing has considerably larger initial cost, its debatable whether eventually the effectiveness of query results outweight the initial costs.

7 References

1. Jamie Callan, Margaret Connell. Query Based Sampling of Text Databases.
2. Milad Shokouhi, Falk Scholer, and Justin Zobel.Sample Sizes for Query Probing in Uncooperative Distributed Information Retrieval.
3. B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. Inf. Process. Manage., 36(2):207227, 2000.