

# Obtaining Language Models of Web Collections Using Query-Based Sampling Techniques

Gary A. Monroe    James C. French    Allison L. Powell  
Department of Computer Science \*  
University of Virginia  
Charlottesville, VA  
{gam9r | french | alp4g}@cs.virginia.edu

## Abstract

*In the context of information retrieval, traditional collection selection algorithms have been widely studied. These algorithms utilize language models, a representation of the contents of each text collection over which selection is to be performed, but these language models cannot always be easily acquired. Query-based sampling is a technique by which these language models are discovered by interacting with a collection and observing the results. Previous work has shown query-based sampling to be a viable solution to the problem of discovering the contents of text collections when the information cannot be otherwise obtained. However, the characteristics of language models of WWW collections created using query-based sampling have not yet been studied. This work evaluates two query-based sampling techniques for building language models of three World Wide Web collections. Experimental results support the effectiveness of query-based sampling as a solution for building language models of web collections. This work also proposes a metric by which it may be possible to determine the point at which further sampling of a given web collection can cease. This metric is used along with other metrics used in previous work to determine the fidelity of these language models.*

## 1 Introduction

The Internet and in particular the World Wide Web has revealed to the public many repositories of information. With many of these collections available for search, the first problem people face when looking for information is trying to decide which collections out of the many to search. When there are only a small number of collections available, searchers can either search all of the collections for

the desired information or they can simply familiarize themselves with the general contents of each collection and direct their queries to one or more specific locations. However, when there are hundreds or thousands of searchable collections and the contents of those collections cannot be easily ascertained without exhaustive measures, automatic collection selection algorithms can be used to assist in the choice of which collections to search by identifying those collections that are most likely to satisfy the information need [2, 3, 4, 6, 8, 11, 12, 13, 14].

Collection selection algorithms require information about the contents of the collections among which they are selecting. We will use the terminology of Callan *et al.* [1] and refer to the summary content information as a language model of the collection. For our purposes, a language model is simply a list of the words that occur in a collection and their frequency of occurrence. Because a completely accurate language model must include a representation of all of the words from all of the documents in a collection, access to all of the documents in the collection is required to generate this complete or base language model. Approaches such as STARTS [5] and Lightweight Probes [7] acquire information from actively participating collections. Because many collection administrators do not provide this level of access to their collections (especially web collections), methods have been proposed for acquiring language models automatically and without the need for cooperation from the other party. One such method is called query-based sampling and is the technique used in the research presented in this paper. Query-based sampling is a sampling technique in which metadata is inferred by interacting with each collection and observing the outcomes [1].

Previous research has been done on query-based sampling to investigate the generality and behavior of the technique under a variety of conditions. Prior research by [1] demonstrated the technique's effectiveness at learning accurate language models for several research testbeds of vary-

---

\*This work supported in part by DARPA contract N66001-97-C-8542 and NASA Grant NAG5-8585.

ing size and heterogeneity. These results were promising but it was not clear if these results could be generalized to World Wide Web collections. A brief study by Monroe *et al.* [9] examined the generality of query-based sampling by examining its effectiveness for generating accurate language models of a collection of web data.

## 2 Research Questions

Prior research has shown that query-based sampling produced collection language models that correlated highly with the actual language models for those collections [1]. A small study has also shown that in one case, query-based sampling produced language models that correlated highly with the actual language models for a collection of web data [9]. However because the prior study with web data only involved one small collection, it is not known if the results of that study are characteristic of other collections of web data with all of the idiosyncrasies of web documents. This previous study also utilized a search engine with a particular size bias in the experiments. It is also unknown what effect, if any, this size bias introduces into the evolution of language models generated from query-based sampling techniques.

The study presented here was intended to address questions about the generality of query-based sampling to various kinds of collections, specifically web collections. The purpose of this research was to study the evolution of language models of web collections and to determine if a learned language model correlated highly to the actual web collection language model could be discovered in a reasonable sample size. An additional goal of this study was to determine if observations made in previous research of web data were generalizable to other collections of web data.

## 3 Experimental Methodology

In this study, we compared three different sampling strategies for the generation of learned language models of three World Wide Web collections. A specific objective of the research was to compare the rate of convergence of the three sampling strategies to a representation of the collections that was “good enough”. Prior work evaluated the fidelity of learned language to the actual language models using two methods:  $ctf$ <sup>1</sup> ratio and the percentage of vocabulary learned [1]. Both of these measures were used here but both require full knowledge of the actual language model. Because such information cannot be known for most real world instances of web collections, an additional measure, the  $df = 1$  ratio<sup>2</sup> [9], was used which can be calculated for

<sup>1</sup>The  $ctf$  is the collection term frequency, the number of occurrences of each term in all documents in a collection.

<sup>2</sup>Document frequency ( $df$ ) is a count of the number of documents containing at least one instance of a term.  $df = 1$  denotes terms that appear in only one document.

an instance in the evolution of the learned language model and compared against its previous snapshot.

### 3.1 Data

Research was conducted on three local collections of web data. Each collection was created from data provided by the DMOZ Open Directory Project<sup>3</sup> (ODP). The open directory project is an effort to create a comprehensive directory of the WWW by employing the help of thousands of volunteer editors. The directory is organized into a category structure. Some search engines use the ODP categories to create their browsing structure. An ODP subdirectory is analogous to a search engine category. DMOZ provides a comprehensive list of the URLs that have been assigned to each individual subdirectory.

Our first collection was created from the list of URLs in the DMOZ/Recreation/Autos subdirectory. We attempted to retrieve and store all of the over 6600 listed pages; some pages were not accessible, resulting in an *Autos* collection of 6,457 HTML files. Similar procedures yielded two more collections. A *Social Sciences* collection was created with 10,065 HTML files, and a *Literature* collection was created with 18,570 HTML files.

The complete or base language models were then built from the collection of locally stored pages using LAMB (LAngeage Model Builder) version 1.3 [10], the mechanism used to build all language models in this study. We used LAMB to fetch each page and extract the vocabulary. All HTML tags except for META tags were parsed out of the pages, and pages that were less than 100 bytes in size were discarded<sup>4</sup>. We removed articles and other very common terms then removed word endings using a conventional stop list and stemming algorithm. The resulting set of terms that were extracted from each page was added to the accumulating learned vocabulary and the term frequency ( $tf$ ) information was updated<sup>5</sup>. The resulting base language model of *Autos* had vocabulary of 37,710 terms, 58% of which had a document frequency of 1. The resulting base language model of *Social Sciences* had vocabulary of 192,401 terms, 63% of which had a  $df$  of 1. The resulting base language model of *Literature* had vocabulary of 212,084 terms, 59% of which had a  $df$  of 1.

### 3.2 Search Engines

Once the three local web collections were created to represent actual web collections, a local search engine was required that would enable the indexing and searching of the local collections and model the behavior of an actual search

<sup>3</sup>More information is available at <http://www.dmoz.org>.

<sup>4</sup>These pages tend to be framesets or error messages and do not contain useful content.

<sup>5</sup>The term frequency is the number of times that particular term occurs in a document.

engine. MiniSearch v.0.2, a search engine freely available on the web<sup>6</sup>, was used as one of the local search engines to conduct the experiments. An investigation of MiniSearch revealed a particular size bias. When presented with a single term query, MiniSearch will retrieve all documents that contain that term and return to the user a result list ordered by the  $tf$  (term frequency) of that term in each file. It follows that, in general, a larger file will be more likely to have a greater  $tf$  for a given term than a smaller file because there is more text. This size bias of MiniSearch compelled us to look for and employ an additional search engine and compare the effects of varying between the two.

Excite's EWS<sup>7</sup> version 1.1.p1 (Excite for Web Servers) was used as the second search engine. This software was also able to be run locally over a set of locally maintained files. Because of proprietary concerns, a detailed investigation of Excite's retrieval algorithm was not possible but presumably the engine includes some type of file size normalization unlike MiniSearch.

### 3.3 Sampling Strategies

Three different sampling strategies were used in these experiments. Two of the strategies were variations on the query-based sampling technique [1]. In query-based sampling, single-term queries are used to acquire samples of the documents in a collection. We retrieved up to 10 documents per query. The third strategy was a random sampling of documents.

For each sampling strategy, 10+ different learned language models were built from 10+ separate runs. The results of all runs were averaged. Both query-based sampling approaches used an initial seed query to begin sampling. The initial seed queries used for the runs of the two query-based sampling approaches were chosen at random from the complete collection language model.

#### 3.3.1 Query-Based Sampling (QBS)

After the top 10 results were returned from an initial seed query, the vocabulary from the result pages was accumulated. One term was chosen at random from the learned vocabulary and became the next query. The results from that query were returned and the vocabulary and term statistics from those pages was accumulated into the learned language model. The next query term was then chosen at random from the updated learned vocabulary. After every approximately 30 pages (Range:[30-39]) had been seen, a snapshot of the evolving language model was saved. This process was continued until the search engine had returned a pre-defined number of unique pages.

<sup>6</sup>More information on MiniSearch is available at <http://www.dansteinman.com>.

<sup>7</sup><http://www.excite.com/navigate>

#### 3.3.2 Query-Based Sampling (no $df = 1$ query terms)

This strategy is identical to the Query-Based Sampling strategy for the first approximately 25 pages returned by the search engine. After about 25 pages are seen, if the randomly chosen term to be used for the next query has a  $df = 1$ , then it was discarded and another term was chosen at random. After every approximately 30 pages (Range:[30-39]) were seen, a snapshot of the evolving language model was saved. This sampling strategy would also continue until the search engine had returned a pre-defined number of unique pages.

#### 3.3.3 Random Document Sampling

For random document sampling, each document had an equal probability of being chosen at each step. One page was chosen at random without replacement from the page list used to build the actual language model. The vocabulary from each page was accumulated into the evolving language model. After every 30 pages seen, a snapshot of the evolving language model was saved. This sampling strategy continued until all pages in the page list were chosen. This strategy models random sampling and provides us with a basis for statistical analysis.

## 4 Analysis

Figures 1, 2, and 3 show that the query-based sampling techniques build language models quickly. Early in the evolution of all three collection language models, the vocabulary grows quickly. The growth slows later in their evolution but never slows to the point where an insignificant number of terms are being added from one snapshot to the next.

Moreover, Figure 1 shows that after 1500 pages have been sampled, 23% of the collection, more than half of the terms in the complete language model have been seen at least once. Figures 2 and 3 represent smaller samples relative to the number of pages that make up their complete models. These plots show, in step with intuition, that at only 16% and 9% of pages seen in *Social Sciences* and *Literature* respectively, many terms have yet to be discovered. Note that the models created by random document selection grow at a slower rate in all three cases. This observation is consistent with previous research and is likely due to the search engine size bias discussed earlier. It is not only MiniSearch which displays this propensity for larger files, even Excite's search engine grows models faster than random document selection.

Another interesting observation is the large difference in vocabulary size between the models of the difference collections. *Autos* with 6,457 pages produces a vocabulary of 37,710 terms, while *Social Sciences*'s 10,065 pages and *Literature*'s 18,570 pages produce vocabularies of 192,401 and

212,084 terms respectively. While the collection sizes increase roughly by a factor of two, the vocabulary sizes do not increase by the same proportion. This attribute of the data could be caused by a variety of factors. One cause might be the heterogeneity of subjects encompassed by a *Literature* or a *Social Sciences* category on the web as opposed to an *Autos* category, which intuitively would seem to be a very specific subject containing similar documents. In addition, *Literature* and *Social Sciences* collections may contain some very large documents like narratives and essays whereas an *Autos* category would likely contain short documents like advertisements and automobile specifications. Whatever the reason, it could be concluded from this data that collection content not collection size may be the primary factor that determines the vocabulary size of a collection.

#### 4.1 *ctf* Ratio

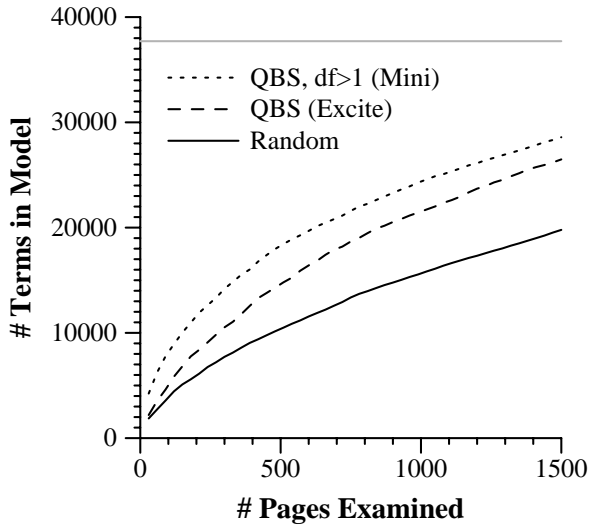


Figure 1. *Autos*, Pages Sampled vs. Terms in Model

Callan *et al.* [1], introduced the *ctf* ratio metric as a way to measure the quality of a collection by weighting the importance of each term. The *ctf* ratio attempts to assess how much of the frequently occurring vocabulary is present. The ratio is defined as follows:

$$\frac{\sum_{i \in LM'} ctf_i}{\sum_{i \in LM} ctf_i}$$

where  $ctf_i$  is the number of times term  $i$  occurs in the collection;  $LM'$  denotes the sampled language model and  $LM$  denotes the complete language model. As figures 4, 5, and 6 show, the *ctf* ratio increases rapidly in all three cases, and at the point when 500 pages have been seen, all cases shown a

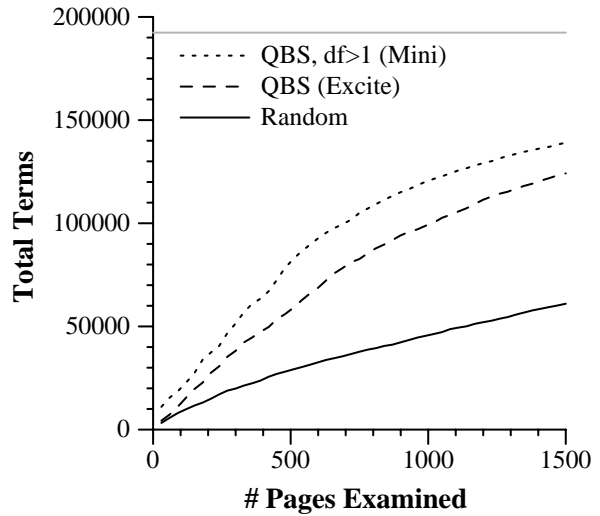


Figure 2. *Social Sciences*, Pages Sampled vs. Terms in Model

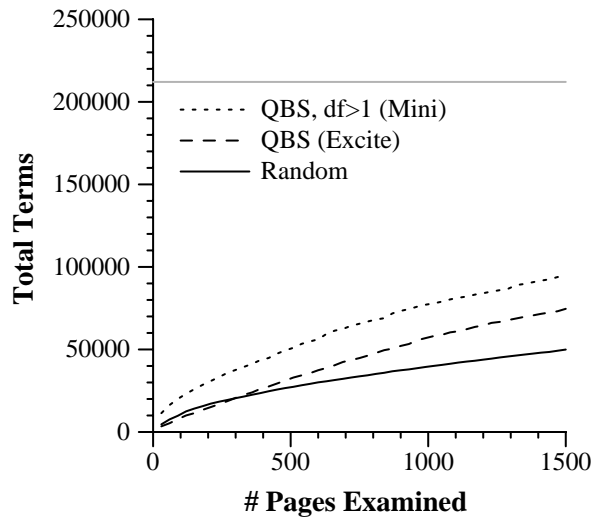


Figure 3. *Literature*, Pages Sampled vs. Terms in Model

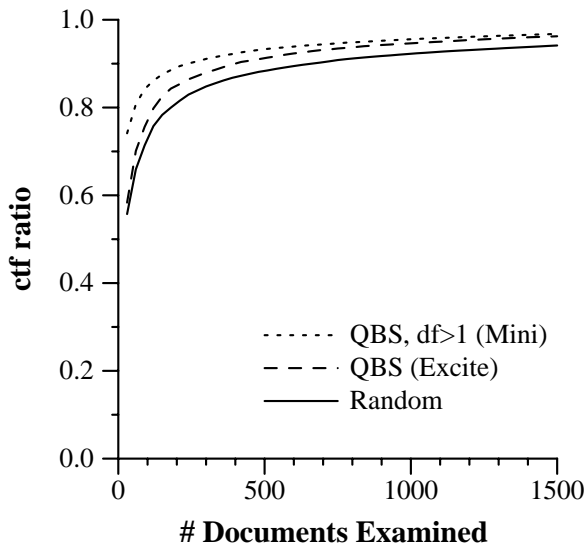


Figure 4. Autos, Number of Documents vs. *ctf* Ratio

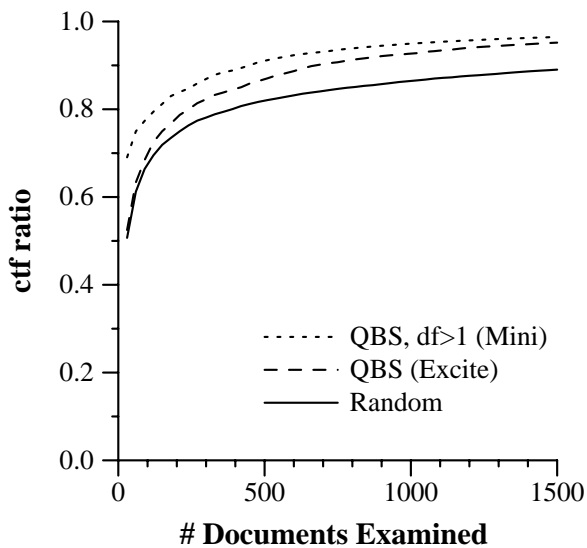


Figure 5. Social Sciences, Number of Documents vs. *ctf* Ratio

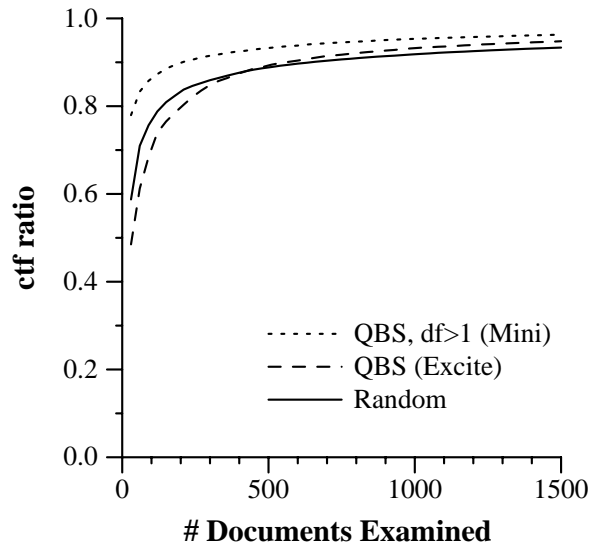


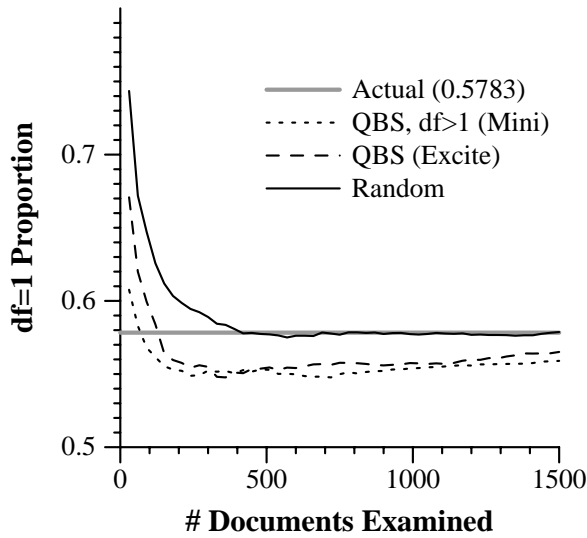
Figure 6. Literature, Number of Documents vs. *ctf* Ratio

*ctf* ratio of over 80%. This indicates that for all three collections, the frequent terms are learned very early in the evolution of the language models. Unfortunately, to calculate the *ctf* ratio, the collection term frequencies are needed. This information is not likely to be available in a real-world situation.

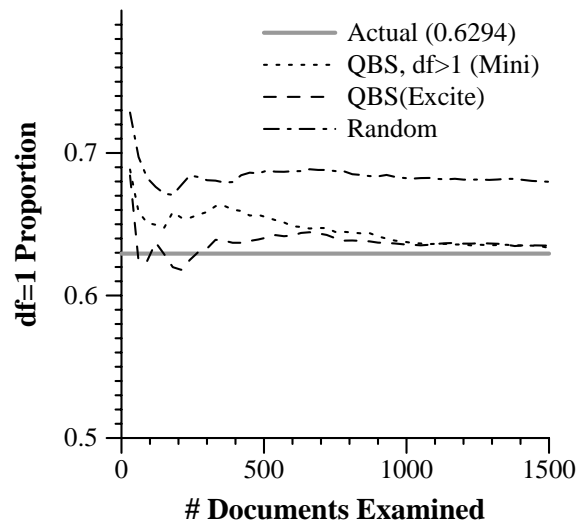
#### 4.2 $df = 1$ Proportion

The  $df1$  proportion metric [9] provides an estimator of how prevalent very rare terms are in the language model. The  $df1$  proportion measures the proportion of terms in the model that have only occurred on one page. This metric does not require knowledge of the complete language model but the complete model does provide information about the value to which the  $df1$  proportion should converge. Figure 7, shows that all three sampling techniques overestimated the  $df1$  proportion early in the model growth. Then after undershooting the proportion, all plots began to converge on the actual value. An objective of this research was to investigate if this behavior could be generalized to other collections of web data. Figures 8 and 9 show the behavior of the  $df1$  proportion during the evolution of language models for *Social Sciences* and *Literature*.

Figure 9, when compared to Figure 7, suggests a similarity between the behavior of the  $df1$  proportion during the evolution of language models of *Autos* and *Literature*. However, Figure 8 shows a very different behavior occurring in the language models of *Social Sciences*. In *Social Sciences* case, the  $df1$  proportion is never really underestimated as it is in the other two cases. The proportions



**Figure 7. Autos, Number of Documents Sampled vs. Proportion of Terms With  $df = 1$**

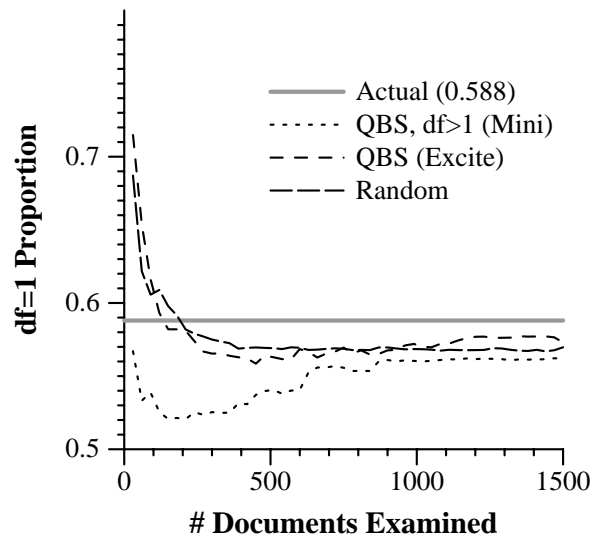


**Figure 8. Social Sciences, Number of Documents Sampled vs. Proportion of Terms With  $df = 1$**

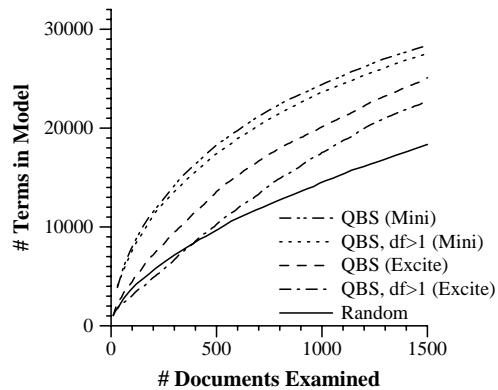
do converge to the actual value as they must but not uniformly. The random document selection sampling strategy produces a language model with a far less accurate  $df1$  proportion early in its evolution than the query-based sampling strategies. These findings are curious and would prompt further research to be conducted with more collections to determine how valuable if at all the  $df1$  proportion metric can be. Some constants, however, can be observed from the  $df1$  proportion metric. There is always a great deal of fluctuation in the beginning. This fluctuation probably can be characterized as start-up behavior and ignored. After this start-up behavior and before convergence, in all cases, the random document selection yields a greater  $df1$  proportion whether the proportion is being overestimated or underestimated. It should also be noted that the actual  $df1$  proportion in all three cases is approximately 60%, which is consistent with observations made in previous work.

### 4.3 Search Engine Differences

Figure 10 is included to give some insight into the performance difference of the two search engines used and the difference introduced by the different versions of the language model builder LAMB. The later version of LAMB, used in all of the results presented earlier, includes terms from META tags along with the content text of the HTML file. A comparison of Figures 10 and 1 show that including terms from META tags increases the vocabulary size slightly. Figure 10 is representative of the behavior in all three collections therefore only one plot was included in this document. The MiniSearch engine consistently performed better than



**Figure 9. Literature, Number of Documents Sampled vs. Proportion of Terms With  $df = 1$**



**Figure 10.** Autos, built from previous version of LAMB, Pages Sampled vs. Terms in Model

the Excite search engine in building language models. This difference can be attributed again to MiniSearch’s size bias. Larger files contain more terms and more terms are better when building a representation of an underlying collection. Figure 10 also displays the performance difference of the sampling strategies. When using query-based sampling techniques, if everything remains constant except for the sampling strategy, choosing query terms based on the term frequency either produces a worse result than if that information was not used or makes no appreciable difference at all.

## 5 Conclusion and Discussion

When using query-based sampling techniques to build language models, the question of when further sampling is no longer necessary is still an open question. However, from observations in this work and other research, a general rule may be to stop when 500 pages have been sampled. Our results suggest that after around 500 pages of a collection are seen, there is enough information to produce a “good enough” representation of the underlying collection for use in collection selection algorithms. This research also leads to the conclusion that collection content has a far greater impact on the representation of the collection than the size of the collection. Some conclusions can also be drawn about the sampling strategies used in this experiment. When performing query-based sampling, the selection of a query term based on the term’s frequency does not improve the terms per document seen rate at which language models are built. The selection of queries based on the term’s frequency can, however, improve the efficiency of the language model builder.

## References

- [1] J. Callan, M. Connell, and A. Du. Automatic Discovery of Language Models for Text Databases. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 479–490, 1999.
- [2] J. P. Callan, Z. Lu, and W. B. Croft. Searching Distributed Collections with Inference Networks. In *Proc. ACM SIGIR Conf. on Research and Devel. in Info. Retrieval*, pages 21–28, 1995.
- [3] N. Craswell, P. Bailey, and D. Hawking. Server Selection on the World Wide Web. In *Proc. ACM Conference on Digital Libraries*, pages 37–46, 2000.
- [4] J. C. French, A. L. Powell, J. Callan, C. L. Viles, T. Emmitt, K. J. Prey, and Y. Mou. Comparing the Performance of Database Selection Algorithms. In *Proc. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pages 238–245, 1999.
- [5] L. Gravano, C.-C. K. Chang, H. García-Molina, and A. Paepcke. STARTS: Stanford Proposal for Internet Meta-Searching. In *Proc. ACM SIGMOD International Conf. on Management of Data*, pages 207–218, May 1997.
- [6] L. Gravano and H. García-Molina. Generalizing GLOSS to Vector-Space Databases and Broker Hierarchies. In *Proceedings of the 21st VLDB Conference*, pages 78–89, 1995.
- [7] D. Hawking and P. Thistlewaite. Methods for Information Server Selection. *ACM Trans. on Information Systems*, 17(1):40–76, 1999.
- [8] W. Meng, K.-L. Liu, C. Yu, X. Wang, Y. Chang, and N. Rishe. Determining Text Databases to Search in the Internet. In *Proceedings of the 24th VLDB Conference*, pages 14–25, 1998.
- [9] G. A. Monroe, D. R. Mikesell, and J. C. French. Determining Stopping Criteria in the Generation of Web-Derived Language Models. Technical Report CS-2000-30, Dept. of Comp. Sci., Univ. of Virginia, May 2000.
- [10] E. K. O’Neil and J. C. French. A Description of the LAMB Web-Derived Language Model Builder. Technical Report CS-2000-31, Dept. of Computer Science, Univ. of Virginia, May 2000.
- [11] A. Sugiura and O. Etzioni. Query Routing for Web Search Engines: Architecture and Experiments. In *Proc. 9th WWW Conf.*, 2000.
- [12] J. Xu and W. B. Croft. Cluster-based Language Models for Distributed Retrieval. In *Proc. ACM SIGIR Conf. on Research and Devel. in Info. Retrieval*, pages 254–261, 1999.
- [13] C. Yu, W. Meng, K.-L. Liu, W. Wu, and N. Rishe. Efficient and Effective MetaSearch for a Large Number of Text Databases. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 217–224, November 1999.
- [14] B. Yuwono and D. L. Lee. Server Ranking for Distributed Text Retrieval Systems on Internet. In *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications*, pages 41–49, April 1997.