IDIAP RESEARCH REPORT

Making Retrieval Faster Through Document Clustering

David Grangier [1] Alessandro Vinciarelli [2]

IDIAP–RR 04-02

- 33 -

[1] IDIAP, CP 592, 1920 Martigny, Switzerland, grangier@idiap.ch
[2] IDIAP, CP 592, 1920 Martigny, Switzerland, vincia@idiap.ch

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 04-02

Making Retrieval Faster Through Document Clustering

David Grangier Alessandro Vinciarelli

Abstract. This work addresses the problem of reducing the time between query submission and results output in a retrieval system. The goal is achieved by considering only a database fraction as small as possible during the retrieval process. Our approach is based on a new clustering technique and comparisons with other clustering methods presented in the literature are performed. Our algorithm is shown to outperform the other techniques: retrieval performances close to those obtained with the whole corpus are achieved by selecting only 5% of the data.

1 Introduction

As the size of document collections grows, there is a need for improving the efficiency of Information Retrieval (IR) systems [11]. The computation time spent between the submission of a query and the output of the results is important as it is directly perceived by the user. Most of such time is required for the computation of the Retrieval Status Value (RSV) of each document. So far, the efforts to reduce this time have mostly focused on data structures (e.g. inverted index files) allowing a faster computation [4]. In this work, we present an approach to further improve efficiency: we limit the number of documents to be searched by selecting a fraction as small as possible of the database without significantly degrading the retrieval performance.

To achieve this goal, the database is first partitioned into disjoint subsets through a clustering algorithm. When a query is submitted, the clusters are ranked according to their matching with it. The better matching clusters are more likely to contain relevant documents than others, thus a good retrieval performance can be achieved by searching only through the documents belonging to them (e.g. it is shown that no degradation in average precision is observed while using only 10% of the database documents). The clustering is performed only once for a given database and it is query independent.

An effective partition must be such that, given a query, most of the relevant documents are concentrated in few clusters accounting for a small fraction of the whole corpus. In this way, retrieval performances similar to those obtained with the whole database can be achieved by examining only few of its documents. Based on the *cluster hypothesis* [10] (closely associated documents tend to be relevant to the same queries), such a partition can be obtained by clustering the documents according to their physical properties (e.g. their term frequencies). We propose a clustering approach where documents are considered similar when their vectors in the $ntf.idf$ (see section 2) space have similar directions. Our method is compared with two other techniques where the similarity is based on different criteria: the first takes into account the number of common terms, the second uses the Kullback-Leibler distance between term distributions [7].

The effectiveness of the three clustering approaches is evaluated by measuring the percentage of relevant documents preserved (the Recall) as a function of the *selection rate* σ (percentage of the corpus represented by the selected documents). The impact of the clustering techniques on the final retrieval performance is also measured (using average Precision, Break Even Point and Precision at top 30 documents) at different selection rates. This allows the comparison of the different approaches at both document selection and retrieval steps. The results show that our approach outperforms the other techniques allowing to achieve higher retrieval performance at any selection rate.

The rest of this paper is organized as follows: section 2 presents the clustering based document selection, section 3 shows our document retrieval approach, section 4 describes experiments and results and section 5 draws some conclusions.

2 Clustering Based Document Selection

The goal of the clustering based document selection is to limit the number of documents to be matched with a given query while achieving performances similar to those obtained with the whole database. To obtain such a result, the collection is split into K disjoint subsets, with $K << D$ (D is the number of documents in the collection), through a clustering algorithm. For a given database, the clustering is performed once and it is query-independent. Each cluster is represented with a single vector and the retrieval process is then composed of two steps: first, the K clusters are ranked according to the matching of their vectors with the query (this step requires to compute only K RSVs). Second, only the documents belonging to the top-ranking clusters are ranked according to their matching with the query.

The rest of this section is organized as follows. Section 2.1 presents the clustering algorithms and section 2.2 shows how document selection is performed.

2.1 Clustering

This step takes as input a collection of documents and splits it into K clusters. The value of K must be small enough to keep the computational cost of the cluster ranking low ($K << D$). On the other hand, if K is too small, the clusters will contain, on average, a large fraction of the corpus: this does not allow one to obtain low selection rates. The value of K must thus be a trade-off between the above effects. The partition of the corpus is effective if the documents relevant to a query tend to concentrate in few clusters. In this way, high Recall can be achieved by selecting few of them (this corresponds, on average, to a low selection rate) and the retrieval performance is not significantly degraded with respect to the use of the whole corpus. According to the cluster hypothesis [10], similar documents tend to be relevant to the same queries. For this reason we based our clustering approach on the similarity of physical properties [6] [9].
Depending on the physical properties used to cluster the documents, the results are different (see section 4). In the approach we propose, we consider similar the documents composed by similar sets of terms and we give more weight to longer documents since we suppose they offer more robust statistics about their term distribution. Our approach is compared with two other techniques: the first simply takes into account the amount of terms shared by the documents, the second one is based on the Kullback-Leibler distance between the term distributions extracted from documents and clusters. As a baseline we used a random partition where the database is split into K subsets containing the same number of documents. The improvement obtained with respect to such a method is an important measure of the effectiveness of the clustering.
All approaches initialize the clusters with a random process (the database is partitioned into K clusters containing the same number of documents) and then perform an iterative refinement until no improvement is observed in the document selection on a set of training queries (the document selection performance is evaluated by measuring the Recall as a function of the selection rate, as shown in section 2.2). The iterative refinement requires to compute the similarity between documents and clusters: each document is assigned to the most similar cluster. In the following, the techniques used to perform such task in the three approaches we compare are described in detail.
In the approach we propose, the similarity between a document d and a cluster C is computed with the inner product between the document vector $\mathbf{d}$ and the cluster vector $\mathbf{c}$:

$$\begin{aligned} sim(d,C) &= \mathbf{d} \cdot \mathbf{c} \\ &= \sum_t d_t \cdot c_t \end{aligned}$$

The document vector $\mathbf{d}$ is calculated as follows:

$$\forall t, d_t = ntf_{d,t} \cdot idf_t$$

where, $ntf_{d,t}$ is the normalized term frequency of term t in document d (the number of occurrences of t in d divided by the total number of term occurrences in d) and idf_t is the inverse document frequency of term t ($idf_t = log(N/N_t)$, where N_t is the the number of documents that contain t and N is the total number of documents).
The vector $\mathbf{c}$ representing cluster C is the barycenter of the vectors representing the documents of C, weighted by their document length:

$$\mathbf{c} = \frac{1}{\sum_{d \in C} l_d} \sum_{d \in C} l_d \cdot \mathbf{d}$$

where l_d is the length of d (i.e. the total number of term occurrences in d). The longer documents are considered more representative of the cluster content and more reliable from a statistical point of view.
In order to have a more complete evaluation of our approach, we compare it with two alternatives proposed in the literature [2][12].

The first method is based on the Kullback-Leibler distance between the term distribution in document d and the term distribution in the set $C^* = \{C, \{d\}\}$ [12]:

$$KL(d, C^*) = \sum_{t:tf_{d,t} \neq 0} p_{t,d} \cdot log \frac{p_{t,d}}{p_{t,C^*}}$$

where

$$p_{t,d} = \frac{tf_{d,t}}{\sum_{t'} tf_{d,t'}}$$

and

$$p_{t,C^*} = \frac{tf_{C,t} + tf_{d,t}}{\sum_{t'}(tf_{C,t'} + tf_{d,t'})}$$

where the term frequency $tf_{C,t}$ in a cluster is defined as the sum of the term frequencies in the documents belonging to it: $tf_{C,t} = \sum_{d \in C} tf_{d,t}$.
The second method, originally introduced in the context of database selection [2], considers the document d as a query and the clusters as documents in a collection. The document is assigned to a cluster according to the following matching measure:

$$sim(d, C) = \sum_{t \in d} ndf_{C,t} \cdot icf_t$$

where $ndf_{c,t}$ is the normalized document frequency of t in C (i.e. the number of documents in C that contain term t divided by the number of documents of C) and $icf_t = log(K/K_t)$ is the inverse cluster frequency (K is the number of clusters and K_t is the number of clusters in which term t is present).
These three clustering approaches used to partition the database are referred to as $ntf \cdot idf$, KL and $ndf \cdot icf$ respectively in the following.

2.2 Document Selection

The goal of this step is to select, given a query, a fraction as small as possible of the database while preserving most of the relevant documents. Such task is performed by ranking the clusters according to their matching with the query. Clusters that contain relevant documents are expected to rank higher than those that do not. Hence, the selection of the n top-ranking clusters should allow one to preserve most of the relevant documents while achieving a low selection rate σ (n is fixed so that the top n clusters contain $\sim \sigma$ percent of all documents).
To perform the ranking, each cluster C is given a vector representation $\mathbf{c} = (c_1, \ldots, c_T)$:

$$\forall t, c_t = ntf_{C,t} \cdot icf_t$$

where the normalized term frequency of t in C is defined as the sum of the term frequencies of the documents of C divided by the sum of the lengths of the documents of C: $ntf_{C,t} = \sum_{d \in C} tf_{d,t} / \sum_{d \in C} l_d$.
The query/cluster matching measure is obtained by projecting the cluster vector along the query direction (the query is represented with a binary vector, i.e. $q_t = 1$ if t is present and 0 otherwise):

$$\begin{aligned} sim(q, C) &= \mathbf{q} \cdot \mathbf{c} \\ &= \sum_{t \in q} ntf_{c,t} \cdot icf_t. \end{aligned} \quad (1)$$

The clusters where the documents contain query terms are considered more likely to contain relevant documents. The document selection step is expected to select most of the relevant documents at a low selection rate. For this reason, we use as a performance measure the Recall (percentage of relevant documents selected) as a function of the selection rate (see 4.1).

3 Document Retrieval

The document retrieval step is based on the Vector Space Model [1] and the Okapi formula is used [8]. Given a document d and a query q,

$$RSV(q,d) = \sum_{t \in q} \frac{(k+1) \cdot tf_{d,t} \cdot idf_t}{k \cdot (1 - b + b \cdot ndl_d) + tf_{d,t}}$$

where ndl_d is the normalized document length of d (i.e. the number of term occurrences in d divided by the average number of term occurences in a document), and k and b are hyper-parameters tuned on training queries. The RSV is computed only for the selected documents, hence the computational weight is heavily reduced for low selection rates. The evaluation of the document retrieval step will measure how small the selection rate can be to keep the retrieval performances acceptable. For this reason, we measure average Precision, Break-Even Point and Precision at top 30 documents for different selection rates. This allows one to compare the values of these measures with those obtained with the whole database. Afterward, a selection rate value can be chosen according to the retrieval performance required and the efficiency targeted.

4 Experiments and Results

This section describes the experiments we performed and the results we obtained. The database used (TDT2 [3]) is composed of $24,823$ documents, with an average length of 178 words. We used the TREC8 queries for training and the TREC9 queries for testing (each set contains 50 queries, with an average length of 6.3 words) [5]. The set of the documents relevant to a query never represents more than 1.1% of the collection and the task can thus be considered as difficult. The database has been preprocessed and normalized by applying stopping (using a generic English stoplist of 389 words) and stemming (using the Porter algorithm).

The database has been partitioned with the four methods described in section 2.1: $ntf \cdot idf$, KL, $df \cdot icf$ and random. All clustering techniques have been initialized with the same random partition in order to perform rigorous comparisons between them. Each experiment has been repeated 10 times with a different initialization and the corresponding results have been averaged to avoid the bias due to a specific initialization. For each approach, we evaluated both document selection, in terms of Recall versus σ, and document retrieval, in terms of Average Precision (AvgP), Break-Even Point (BEP) and Precision at top 30 documents (P30) for different σ values. In the experiments that follow, we used $K = 800$ (K is the number of clusters). This value is shown to be small enough to improve the efficiency (K is $\sim$30 times smaller than the corpus size) and large enough to limit the retrieval performance loss.

The rest of this section is organized as follows: section 4.1 presents the document selection evaluation and section 4.2 shows the document retrieval performances at different selection rates.

4.1 Document Selection

The document selection performance is measured in terms of Recall as a function of the selection rate (see section 2.2). A high Recall at a low selection rate means that the system is able to select most of the relevant documents even discarding a high percentage of the corpus.

The results (see figure 1) show that the clustering technique we propose ($ntf \cdot idf$) outperforms the others at any σ value. This is especially evident at low selection rates: at σ=5% the Recall is 76.8% for $ntf \cdot idf$, and 68.0%, 42.3% and 36.4% for $df \cdot icf$, KL and *random* respectively. The results of $df \cdot icf$ are close to those obtained with $ntf \cdot idf$, but only at high selection rates (e.g. $R_{df \cdot icf} = 93.1\%$ and $R_{ntf \cdot idf} = 94.1\%$ at $\sigma = 40\%$). With $ntf \cdot idf$, only $\sim$15% of the relevant documents are lost, when the number of document to be searched is divided by 10 ($R_{ntf \cdot idf}$=84.1% at σ=10%). Such a loss is acceptable for many applications (e.g. web-searching) where the user is not necessarily interested to

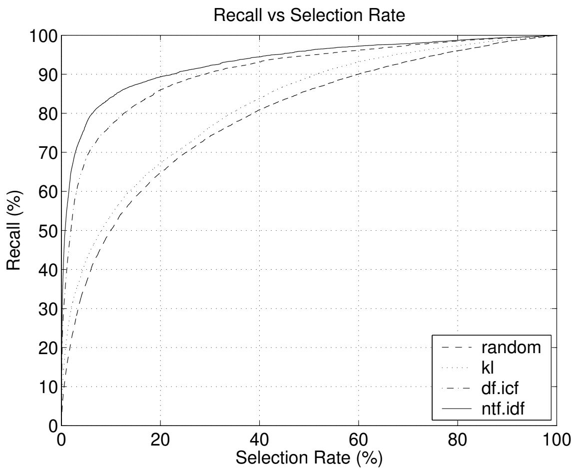

Figure 1: Comparing the document selection performance of the different clustering techniques

the whole set of relevant documents.
The performance achieved by KL is worse than those of the two above techniques. At any selection rate, the results are similar to those obtained with the random partition. This depends, in our opinion, on the fact that term distributions extracted from short documents (148 words, on average) are not statistically reliable. On the contrary, techniques based on term weighting, such as $df \cdot icf$ and $ntf \cdot idf$, are more robust and have been shown to concentrate most of the relevant documents in few clusters.
The results discussed in this section concern the document selection step. It is necessary to measure the impact of the document selection on the retrieval performance. This will also show if a better performance at the document selection step leads to better retrieval results. In this case, the choice of a clustering algorithm could be made without the need of complete retrieval experiments.

4.2 Document Retrieval

Document retrieval is the final step of the process. Only the selected documents are ranked according to their RSV. The performances obtained with the selected documents are compared with those obtained with the whole corpus (see section 3). The comparison is made using standard IR measures (AvgP, BEP or P30). A good partition should allow one to observe moderate performance degradation, even at low selection rates.
The retrieval results (see figure 2) show that, for $ntf \cdot idf$ and $df \cdot icf$, the performances at selection rates higher than 20% are similar to those obtained with the whole corpus (e.g. at σ=20%, avgP is 32.9% and 32.3% for $ntf \cdot idf$ and $df \cdot icf$ repectively, when σ=100%, $AvgP$=32.7%). This means that 80% of the computational cost can be avoided, without observing differences in the retrieval performance.
At low selection rates, there is a limited degradation in P30 for $ntf \cdot idf$: at σ=5%, $P30_{ntf \cdot idf}$=27.1%, whereas $P30$=27.4% is obtained at σ=100%. This means that, even if some relevant documents are lost at low selection rates (at σ=5%, 23% of the relevant documents have not been selected), the impact on the first positions of the ranking is low. This happens because, in our opinion, the relevant documents lost at the selection step are also among those that are ranked poorly when the whole database is considered.
As in the document selection evaluation, $ntf \cdot idf$ outperforms the other methods at any selection rate. The $df \cdot icf$ leads to similar results at selection rates higher than 20% (e.g. $BEP_{ntf \cdot idf}$=34.5% and $BEP_{df \cdot icf}$=34.1% at σ=20%), but, at low selection rates ($\sigma < 10\%$), $ntf \cdot idf$ achieves better results (e.g. at σ=5%, $avgP_{df \cdot icf}$=29.4% and $avgP_{ntf \cdot idf}$= 31.6%). The KL clustering introduces more degradation in the retrieval performance than $df \cdot icf$ and $ntf \cdot idf$, at any selection rate. For

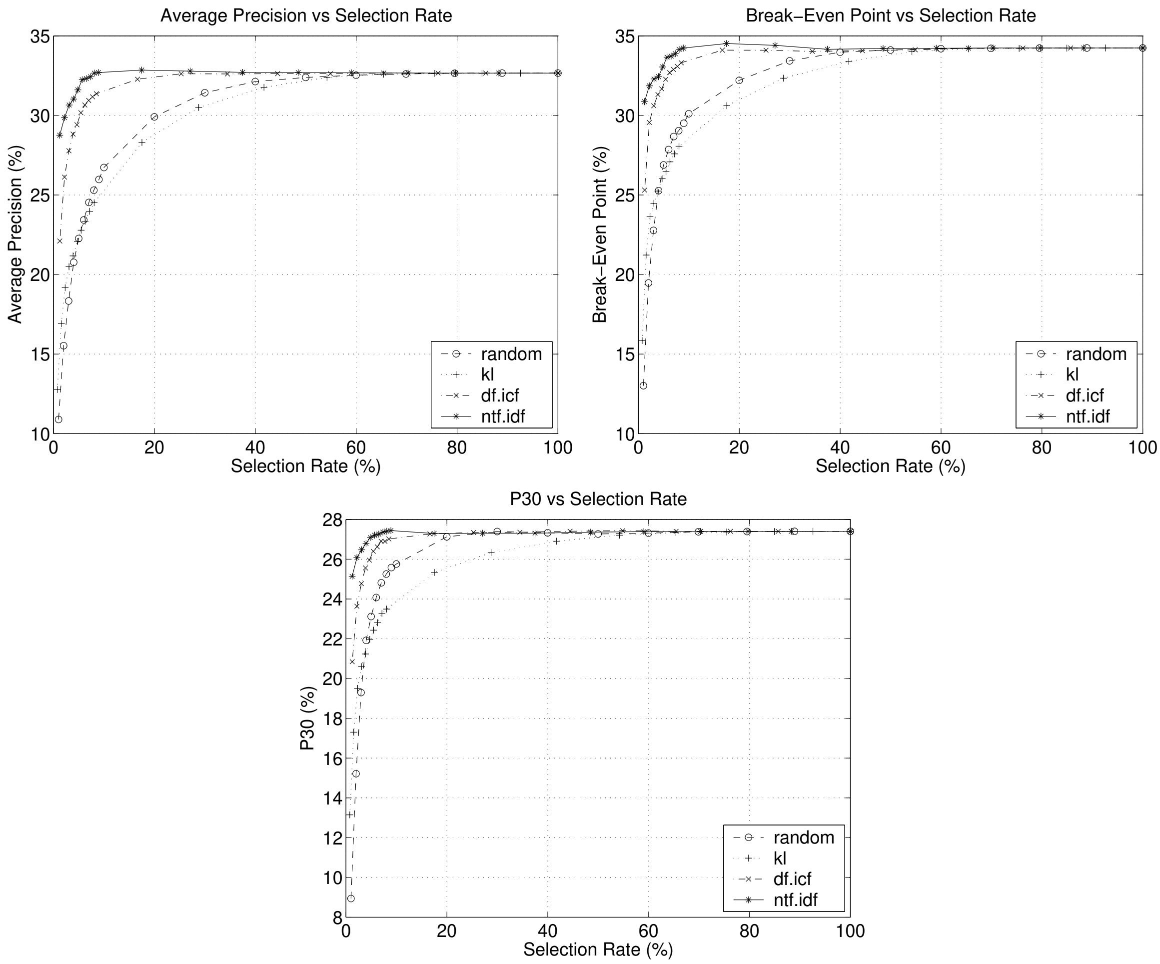

Figure 2: Comparing the document retrieval performance for different selection rates

σ between 5% and 60%, the avgP obtained with KL is even lower than the one obtained with the random partition (e.g. $AvgP_{KL}$=28.3% and $AvgP_{random}$=29.9% at σ=20%).
The results of $df \cdot icf$ and $ntf \cdot idf$ clustering techniques are promising: moderate degradation in the retrieval performance has been observed even when only 5% of the documents are searched. Computation cost can hence be significantly reduced, while having the same number of relevant documents in the top 30 positions of the ranking. This is especially useful for applications in which the user is looking for only few relevant documents in a large database.

5 Conclusions

As the size of the databases grows, there is a need to reduce the computation time spent between the submission of a query and the output of the results. In this paper, we present an approach to reduce this time: the number of documents to be searched is limited to a small fraction of the database.
To achieve this goal, the documents are clustered. When a query is submitted, the clusters are ranked according to their matching with it. Only the documents belonging to the top-ranking clusters are then ranked with the retrieval system. This approach allows one to obtain good retrieval performances if the top-ranking clusters contain most of the relevant documents.

Based on Van Rijsbergen cluster hypothesis, we propose a clustering technique ($ntf \cdot idf$) to group documents according to their physical properties. In this method two documents are considered similar if their vectors are similar in the $ntf \cdot idf$ space. This method has been compared with two other clustering techniques. In the first method ($df \cdot icf$), the document similarity is also based on a term weighting. In the second method (KL), the document similarity is computed by comparing term distributions. The effectiveness of the three algorithms has been compared by measuring the number of relevant documents preserved as a function of the selection rate σ (the fraction of the database to be searched). The impact of the clustering techniques on the retrieval performance (using AvgP, BEP, P30) has also been measured at different selection rates.

The results have shown that our method outperforms the other clustering techniques: at any selection rate, $ntf \cdot idf$ selects more relevant documents (e.g. at σ=10%, 84.1% of the relevant documents are preserved using $ntf \cdot idf$, while respectively 76.9% and 53.9% of the relevant documents are preserved using $df \cdot icf$ and KL). It has also been shown that, for $ntf \cdot idf$, only a little degradation in the retrieval performance is observed, even at low selection rates (e.g. at σ=5%, $AvgP_{ntf \cdot idf}$=31.6%, whereas Avg=32.6% is obtained at σ=100%). The results of the $df \cdot icf$ clustering techniques are close to those of $ntf \cdot idf$, especially at high selection rates. On the contrary, KL clusters have been shown to be ineffective.

In this work, clustering based document selection has been shown to allow good retrieval performance, when searching only through a small fraction of the corpus. However, our results must be confirmed on different databases. As a future work, the use of different techniques to rank clusters should also be investigated.

6 Acknowledgments

This work is supported by the Swiss National Centre Foundation through the National Center of Competence in Research on Interactive Multimodal Information Management (IM2).

References

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Harlow, England, 1999.

[2] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collection with inference networks. In *Proceedings of SIGIR '95 Conference*, pages 21–28. ACM, July 1995.

[3] C. Cieri, D. Graff, M. Liberman, N. Martey, and S. Strassel. The TDT-2 text and speech corpus. In *Proceedings of the DARPA Broadcast News Workshop*, pages 57–60, February 1999.

[4] W. B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Upper Saddle River, New Jersey, 1992.

[5] J.S. Garofolo, G.P. Auzanne, and E.M. Voorhees. The TREC SDR track: A success story. In *Proceedings of Content-Based Multimedia Information Access Conference*, pages 1–20. ELSNET, April 2000.

[6] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 3(2):159–165, 1958.

[7] C. D. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

[8] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference*, pages 109–126, November 1994.

[9] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[10] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[11] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes*. Morgan Kaufmann, San Francisco, California, 1999.

[12] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of SIGIR '99 Conference*, pages 254–261. ACM, August 1999.