Exploring the Cluster Hypothesis, and Cluster-Based Retrieval, over the Web

Fiana Raiber
fiana@tx.technion.ac.il

Oren Kurland
kurland@ie.technion.ac.il

Faculty of Industrial Engineering and Management, Technion
Haifa 32000, Israel

ABSTRACT

We present a study of the *cluster hypothesis*, and of the performance of cluster-based retrieval methods, performed over large scale Web collections. Among the findings we present are (i) the cluster hypothesis can hold, as determined by a specific test, for large scale Web corpora to the same extent it does for newswire corpora; (ii) while spam documents do not affect the extent to which the cluster hypothesis holds, they considerably affect the performance of cluster-based, as well as that of document-based, retrieval methods; and, (iii) as is the case for newswire corpora, cluster-based methods can yield better performance than document-based methods for Web corpora.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

Keywords: cluster hypothesis, cluster-based retrieval

1. INTRODUCTION

The cluster hypothesis states that "closely associated documents tend to be relevant to the same requests" [20]. The hypothesis gave rise to a large body of work on devising cluster-based retrieval methods. However, tests measuring the extent to which the hypothesis holds [6, 21, 4, 18], as well as cluster-based retrieval methods (see [10] for a survey), were employed upon small scale newswire collections that are composed of well edited documents.

The study we present in this paper is, to the best of our knowledge, the first to (i) explore the cluster hypothesis over large scale and noisy Web corpora, (ii) compare cluster-based retrieval methods over such corpora; and, (iii) analyze the effect of spam on the extent to which the cluster hypothesis holds, and on the performance of cluster-based retrieval methods.

The main findings that our extensive empirical exploration reveals are as follows: (i) the cluster hypothesis, as tested using Voorhees' nearest-neighbor test [21], can hold to the same extent for large scale Web corpora as it holds for newswire corpora; (ii) spam has no effect on the extent to which the cluster hypothesis holds, yet it substantially affects the performance of cluster-based and document-based retrieval methods; (iii) cluster-based retrieval methods can often outperform document-based retrieval methods over large scale Web corpora; and, (iv) overlapping clusters can be more effective for cluster-based retrieval than hard clusters for both newswire and Web corpora.

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

2. RELATED WORK

Several tests have been proposed to measure the extent to which the cluster hypothesis holds [6, 21, 4, 18]. We adopt Voorhees' nearest neighbor test [21] that was shown to be correlated with the performance of some cluster-based retrieval methods [16].

Some cluster-based retrieval methods utilize clusters that are created offline from the entire corpus [6, 3, 12, 10]. However, clustering of large scale Web collections is not feasible. Hence, we study the performance of state-of-the-art methods that utilize query-specific clusters [7, 14, 8], that is, clusters created from the documents most highly ranked by some initial search [22]. Query-specific clusters were used in two capacities. First, for ranking documents by first inducing a ranking over clusters and then transforming it to a ranking over documents [12, 9, 13, 23, 7, 14]. Second, for directly ranking documents by using clusters to smooth documents' representations (e.g., language models) [12, 8]. We study the performance of state-of-the-art cluster-based methods that represent these two categories.

3. EMPIRICAL EXPLORATION

The goal of the first set of experiments is examining the extent to which the cluster hypothesis holds in the Web setting. To that end, we use Voorhees' nearest neighbor (NN) test [21].[1] Specifically, an initial list $\mathcal{D}_q^{[n]}$ of n documents is retrieved from corpus $\mathcal{D}$ in response to query q using some search algorithm; a document d is scored in response to q by their language-model-based similarity, denoted $p_d(q)$. Details regarding the estimate $p_y(x)$ of the similarity between texts x and y are provided in Section 3.1. For each relevant document d in $\mathcal{D}_q^{[n]}$, we count the number of relevant documents in $\mathcal{D}_q^{[n]}$ that are among d's $k-1$ nearest neighbors in $\mathcal{D}_q^{[n]}$; $p_{d'}(d)$ is the estimate for the similarity between d and d'. The sum of counts over all the relevant documents found

[1]The nearest neighbor test was shown to be insufficient for testing the cluster hypothesis when a query-biased inter-document similarity measure was employed [18]. In this study we use a query-independent similarity estimate.

for all tested queries is then divided by the total number of the relevant documents.

The goal of the second set of experiments is comparing the effectiveness of several state-of-the-art cluster-based retrieval methods with that of highly effective document-based retrieval approaches. In what follows we provide a short description of the retrieval methods explored.

Let $C(\mathcal{D}_q^{[n]})$ be a set of clusters created from $\mathcal{D}_q^{[n]}$ using some clustering algorithm. All cluster-ranking-based methods that we explore first rank the clusters c ($\in C(\mathcal{D}_q^{[n]})$). Each cluster is then replaced by its constituent documents (while omitting repeats if there are) to induce a (re-)ranking of documents in $\mathcal{D}_q^{[n]}$; within-cluster document ordering is based on the initial documents' retrieval scores [12].

The **ArithMean** cluster-ranking-based method [7, 14] scores cluster c by the arithmetic mean of the initial retrieval scores of its constituent documents: $\frac{1}{|c|}\sum_{d\in c} p_d(q)$; $|c|$ is the number of documents in c. A geometric-mean-based representation of a cluster was shown to be more effective than a range of previously proposed representations [14, 17]. Accordingly, another method that we study is **GeomMean** which assigns cluster c with the score: $\prod_{d\in c} p_d(q)^{\frac{1}{|c|}}$.

ArithMean and GeomMean are based solely on the initial retrieval scores of documents in the clusters. The state-of-the-art **ClustRanker** method [7] incorporates, in addition, measures of document and cluster biases. A cluster c is scored by $\lambda Cent(c)p_c(q) + (1-\lambda)\sum_{d\in c} p_d(q)p_d(c)Cent(d)$; $Cent(d)$ and $Cent(c)$ quantify the centrality of document d with respect to $\mathcal{D}_q^{[n]}$, and that of cluster c with respect to $C(\mathcal{D}_q^{[n]})$, respectively. The centrality is computed using a variant of the PageRank algorithm that utilizes inter-document and inter-cluster similarities.

The aforementioned methods are based on ranking of clusters as a first step. A different approach is that which uses clusters to *directly* rank documents. One such method, interpolation-f [10, 8] (**InterpF** in short), assigns document d ($\in \mathcal{D}_q^{[n]}$) the score $\lambda p_d(q) + (1-\lambda)\sum_{c\in C(\mathcal{D}_q^{[n]})} p_c(q)p_d(c)$.

We compare the performance of the cluster-based retrieval methods specified above with that of two state-of-the-art document-based retrieval approaches. The first is **RM3**, a query expansion method that uses the documents in $\mathcal{D}_q^{[n]}$ to expand the query [11, 1]. The second is the Markov Random Field (**MRF**) method, which utilizes term proximities and which was shown to be highly effective for Web retrieval [15]. RM3 and MRF, as the cluster-based methods, re-rank $\mathcal{D}_q^{[n]}$.

3.1 Experimental setup

Experiments were conducted using a variety of both Web and newswire TREC datasets, specified in Table 1.

corpus	# of docs	data	queries
AP	242,918	Disks 1-3	51-150
TREC8	528,155	Disks 4-5 (-CR)	401-450
WT10G	1,692,096	WT10g	451-550
GOV2	25,205,179	GOV2	701-850
ClueWebA ClueWebAF	503,903,810	ClueWeb09 (Category A)	1-150
ClueWebB ClueWebBF	50,220,423	ClueWeb09 (Category B)	1-150

Table 1: Data used for experiments.

AP and TREC8 are mainly composed of news articles. WT10G is a small noisy Web collection, and GOV2 is a crawl of the .gov domain, and hence contains mostly well edited documents. Both categories A and B of the ClueWeb collection were used (ClueWebA and ClueWebB). Two additional experimental settings for ClueWeb, denoted ClueWebAF and ClueWebBF, were created following previous work [2]. Specifically, highly ranked documents that were assigned by Waterloo's spam classifier with a score below 70 and 50 in categories A and B, respectively, were filtered out until n documents were accumulated, and the initial ranking of the residual corpus was then used.

We applied Krovetz stemming and removed stopwords (on the INQUERY list) from queries but not from documents, via the Indri toolkit (http://www.lemurproject.org/indri/). Titles of TREC topics served as queries.

The language-model-based similarity between texts x and y, $p_y(x)$, is set to $\exp\left(-CE\left(p_x^{[0]}(\cdot)\;||\;p_y^{[\mu]}(\cdot)\right)\right)$, where $p_z^{[\mu]}(\cdot)$ is the Dirichlet-smoothed unigram language model induced from z with the smoothing parameter μ [10]; μ is set to 1000 [24]. As noted above, this measure is used to create the initial ranking of documents with respect to the query (henceforth **Initial**) and to measure the similarities between queries, documents and clusters. A cluster is represented by the concatenation of its constituent documents [12, 7, 8]; the order of concatenation has no effect, because the cluster-based methods use unigram language models that do not consider term proximities.

We use two types of clusters utilized in work on cluster-based retrieval [19, 8]. The first is overlapping k nearest neighbors clusters (**KNN**). A cluster is created for each d ($\in \mathcal{D}_q^{[n]}$), and comprises d and its $k-1$ nearest neighbors d' in $\mathcal{D}_q^{[n]}$ that yield the highest $p_{d'}(d)$. In addition, we use non-overlapping hierarchical agglomerative clusters (**HAC**) based on the complete-link clustering algorithm, wherein clusters are merged bottom-up, and the partition of clusters with the smallest difference between k and the average size of the clusters in the partition is chosen. To create HAC clusters, the dissimilarity between documents d_i and d_j is set to $\frac{1}{p_{d_j}(d_i)} + \frac{1}{p_{d_i}(d_j)}$, which is a symmetric measure.

Unless otherwise specified, the initially retrieved list, $\mathcal{D}_q^{[n]}$, contains $n = 50$ documents. As already noted, all the retrieval methods we study re-rank $\mathcal{D}_q^{[50]}$ so as to improve upon its initial ranking. Accordingly, we use MAP (at cutoff 50) and the precision at the top 5 ranks (P@5) for evaluation metrics. Performance patterns similar to those of P@5 were observed for NDCG@5; these numbers are omitted due to space considerations. Statistically significant differences are determined using the two tailed t-test with $p = 0.05$.

The parameter k, which controls the size of clusters used in the cluster-based retrieval methods, and the cluster hypothesis nearest-neighbor test, is set to 5. The free-parameter values of the various retrieval methods are set for each method, collection and evaluation metric by either (i) selecting the values that optimize average performance over all queries in an experimental setting (**OPT**), or (ii) performing 10-fold cross validation over the queries (**CV**). The first approach is intended for studying the potential performance of the methods when ameliorating the effects of free-parameter values. The goal of using the second approach is evaluating performance when free-parameter values are learned.

The parameter λ in ClustRanker and InterpF is set to a value in $\{0, 0.1, \ldots, 1\}$. The number of nearest neighbors and the dumping factor in the PageRank algorithm used by ClustRanker [7] are set to values in $\{4, 9, 19, 29, 39, 49\}$ and $\{0.05, 0.1, \ldots, 0.95\}$, respectively. We use the sequential dependence model of MRF [15] and select the values of its free parameters from $\{0, 0.05, \ldots, 1\}$. The relevance model, RM3 [1], is constructed from all documents in $\mathcal{D}_q^{[n]}$; the Dirichlet smoothing parameter, the number of terms and the original-query weight are set to 1000, a value in $\{5, 10, 25, 50\}$, and a value in $\{0.1, 0.3, \ldots, 0.9\}$, respectively.

3.2 Experimental results

3.2.1 The cluster hypothesis test

	$n = 50$	$n = 100$	$n = 250$	$n = 500$	$n = 1000$
AP	3.1	3.1	3.0	2.8	2.7
TREC8	2.7	2.6	2.4	2.3	2.2
WT10G	2.5	2.4	2.3	2.1	2.0
GOV2	3.3	3.2	3.0	2.8	2.6
ClueWebA	2.6	2.5	2.3	2.2	2.0
ClueWebAF	2.7	2.6	2.3	2.1	2.0
ClueWebB	2.9	2.7	2.4	2.2	2.0
ClueWebBF	2.9	2.7	2.4	2.2	2.0

Table 2: The cluster hypothesis test. Numbers represent the (average) number of relevant documents among the 4 nearest neighbors of a relevant document; n is the number of documents in the result list, $\mathcal{D}_q^{[n]}$, that is analyzed.

The results of the nearest neighbor cluster hypothesis test, for result lists $\mathcal{D}_q^{[n]}$ of varying sizes n, are presented in Table 2. We can see that the numbers for all the (Web and non-Web) collections decrease with increasing values of n. This finding is consistent with previous reports [19]. The lowest numbers are often observed for WT10G, which is a small noisy Web collection. The highest numbers are reported for GOV2 (for n<1000), which contains well edited documents. For n>100, the numbers for AP, a small newswire collection, are as high as those for GOV2.

Although TREC8 is mainly composed of news articles and is much "cleaner" than ClueWebAF, the numbers for these two collections are identical for small values of n. At first glance, this finding might suggest that the cluster hypothesis, as measured by the nearest neighbor test, holds to the same degree on these two, rather different, collections. However, a closer look at Table 2 reveals that with increasing values of n, the numbers for ClueWebAF are lower than those for TREC8; furthermore, the numbers for ClueWebAF become exactly the same as those for WT10G; both WT10G and ClueWebAF are noisy Web collections.

There are no differences between the numbers for ClueWebB and ClueWebBF, and the differences between the numbers for ClueWebA and ClueWebAF are minor. This finding means that the existence of spam documents in the result list $\mathcal{D}_q^{[n]}$ that is analyzed does not affect the degree to which the cluster hypothesis test holds for these collections.

To summarize, the nearest neighbor cluster hypothesis test holds to a greater extent on a clean Web collection (GOV2) than on noisy Web collections (WT10G and the ClueWeb collections). Furthermore, spam has no noticeable effect on the extent to which the test holds. Finally, in some cases the test might hold to a greater extent for Web collections than for newswire collections.

3.2.2 Cluster-based retrieval

Table 3 presents the performance of the cluster-based and document-based retrieval methods. As a reference comparison we use a method termed **Optimal** which uses a ranking of the clusters induced by the *true* percentage of relevant documents they contain as determined using relevance judgments. The performance of Optimal is by far the best in Table 3. This finding, which is in accordance with previous reports on ranking clusters in newswire domains [5, 19, 13], attests to the potential merit of methods that can detect clusters containing a high percentage of relevant documents.

When using KNN clusters, the cluster-based methods outperform the initial document-based ranking (Initial) in most reference comparisons (corpus × evaluation metric); quite a few of the improvements, specifically for InterpF, are statistically significant. This attests to the overall effectiveness of cluster-based retrieval.

In comparing the arithmetic mean and geometric mean based representations for clusters, we see that in contrast to findings reported for newswire corpora [14, 17], the former is as effective as the latter. Specifically, ArithMean is at least as effective as GeomMean in half of the relevant comparisons for KNN clusters, and in all relevant comparisons (except for a single case of P@5 for AP) for HAC clusters.

We next explore the relative performance patterns of the cluster-based methods. For KNN clusters, the performance of none of the cluster-ranking-based methods ArithMean, GeomMean and ClustRanker dominates that of the other two; recall that these methods rank clusters and then transform the ranking to that of documents. For HAC clusters, ClustRanker often outperforms ArithMean and GeomMean. The InterpF method, which directly ranks documents by utilizing cluster-based information, is more effective in most relevant comparisons than the three cluster-ranking-based methods; the main exception is the comparison with ClustRanker for HAC clusters. This finding implies that using clusters to directly rank documents can be more effective than methods that rely on ranking of clusters.

In comparing the performance of the cluster-based methods when using KNN and HAC clusters, we see that using KNN often results in better retrieval performance. This finding, which is in line with some previous reports for newswire corpora [8], attests to the potential merits of using overlapping clusters with respect to using hard clusters.

As already noted, no (substantial) differences were observed for the nearest neighbor test when applying filtering, or not, of spam documents from the result list. Yet, the retrieval performance of the cluster-based methods, and that of the document-based methods, for the ClueWeb settings is much higher when spam documents are filtered out.

For ClueWebA and ClueWebAF, which are the largest (Web) collections explored, as well as for AP and TREC8, the small scale newswire corpora, the best performance in *all* experimental settings is attained by a cluster-based method. Cluster-based methods also obtain the best performance for P@5 for WT10G and GOV2, and for MAP for ClueWebB. These findings attest to the potential merits of applying cluster-based retrieval, with respect to document-based retrieval, whether the collection is newswire or Web, and whether it is of small scale or large scale.

	AP		TREC8		WT10G		GOV2		ClueWebA		ClueWebAF		ClueWebB		ClueWebBF	
	MAP	P@5	MAP	P@5	MAP	P@5	MAP	P@5	MAP	P@5	MAP	P@5	MAP	P@5	MAP	P@5
Initial	9.2	46.1	16.9	46.8	13.8	33.4	11.7	56.2	2.1	10.1	5.8	31.8	9.9	24.6	12.4	35.9
RM3 (OPT)	9.4	47.3	17.0	46.8	**14.1**	34.8	**12.6**^{i}	60.3^{i}	2.4^{i}	10.9	6.5^{i}	33.8	11.9^{i}	33.8^{i}	**13.8**^{i}	**41.1**^{i}
RM3 (CV)	9.2	46.5	16.8	43.6^{i}	13.3	32.4	12.6^{i}	57.7	2.4^{i}	10.0	6.4^{i}	31.5	11.9^{i}	33.8^{i}	13.8^{i}	41.1^{i}
MRF (OPT)	9.4	47.9	16.9	49.2	14.0	35.7	12.4^{i}	60.9^{i}	3.0^{i}	15.3^{i}	6.3^{i}	33.2	11.4^{i}	30.8^{i}	13.5^{i}	39.2^{i}
MRF (CV)	9.1	45.7	16.8	47.6	13.6	34.6	12.2^{i}	59.2	2.9^{i}	14.3^{i}	6.3^{i}	33.0	11.0^{i}	29.2^{i}	13.4^{i}	38.0
KNN																
Optimal	14.4^{i}	79.4^{i}	23.4^{i}	81.6^{i}	21.5^{i}	66.8^{i}	17.3^{i}	93.2^{i}	5.7^{i}	44.5^{i}	11.0^{i}	65.0^{i}	21.4^{i}	69.6^{i}	22.5^{i}	70.1^{i}
ArithMean	9.8	48.1	16.6	44.8	14.0	**35.9**	12.4^{i}	**61.8**^{i}	2.2	11.6	6.2^{i}	**34.2**	9.5	24.3	12.6	35.0
GeomMean	9.8^{i}	49.5	17.0	46.8	**14.1**	35.5	12.3^{i}	60.4^{i}	2.2	11.1	6.2^{i}	34.1	9.8	25.5	13.0	36.4
ClustRanker (OPT)	9.6	48.7	16.9	48.0	13.4	33.0	12.2	61.2	3.3^{i}	**17.8**^{i}	**6.6**^{i}	32.6	11.9^{i}	**34.3**^{i}	13.3	38.6
ClustRanker (CV)	9.2	42.2	16.7	37.2^{i}	12.5	27.4^{i}	12.0	57.4	2.8^{i}	14.2	6.6^{i}	29.6	11.9^{i}	31.5^{i}	12.5	32.4
InterpF (OPT)	**9.9**^{i}	**51.1**^{i}	**17.1**	49.6	13.9	35.1	12.3^{i}	59.6	3.1^{i}	15.5^{i}	6.4^{i}	32.7	11.6^{i}	33.5^{i}	13.2^{i}	38.8
InterpF (CV)	9.9^{i}	51.1^{i}	17.0	49.6	13.8	33.8	12.2^{i}	57.6	3.1^{i}	15.5^{i}	6.4^{i}	31.8	11.5^{i}	33.5^{i}	13.1^{i}	37.6
HAC																
Optimal	12.9^{i}	69.1^{i}	21.3^{i}	70.8^{i}	18.3^{i}	49.3^{i}	15.8^{i}	82.4^{i}	5.0^{i}	31.1^{i}	9.7^{i}	56.8^{i}	16.9^{i}	53.0^{i}	18.9^{i}	60.4^{i}
ArithMean	9.1	42.0	16.2	44.0	10.6^{i}	26.4^{i}	11.9	51.9^{i}	2.2	8.8	5.3^{i}	25.1^{i}	9.6	20.7^{i}	12.3	31.6^{i}
GeomMean	9.0	42.4	16.1	44.0	9.4^{i}	24.7^{i}	11.4	48.9^{i}	2.1	7.7^{i}	5.3^{i}	25.0^{i}	9.2^{i}	19.1^{i}	12.3	31.6^{i}
ClustRanker (OPT)	9.7	47.3	16.4	**50.8**	12.9	35.1	12.2	56.8	**3.4**^{i}	15.4^{i}	6.5^{i}	33.6	**12.2**^{i}	31.6^{i}	12.9	37.7
ClustRanker (CV)	9.1	43.8	15.4	50.0	12.3	31.1	12.0	54.9	3.4^{i}	14.6^{i}	6.5^{i}	33.6	12.1^{i}	30.8^{i}	12.5	33.8
InterpF (OPT)	9.2	46.3	16.9	46.8	13.8	33.4	11.7	56.2	2.2	11.2	5.8	31.8	9.9	24.6	12.4	35.9
InterpF (CV)	9.1	45.7	16.9	45.6	13.8	33.0	11.7	56.2	2.1	10.3	5.8	31.8	9.9	24.2	12.4	35.9

Table 3: Retrieval performance. (OPT) and (CV) indicate the procedure used to set free-parameter values. The best result (except for that for Optimal) in a column is boldfaced for OPT and underlined for CV. 'i' marks statistically significant differences with Initial. Note: ArithMean and GeomMean do not incorporate free parameters, and hence, there is no difference between OPT and CV for them.

4. CONCLUSIONS

We presented the first empirical study for large scale Web corpora of (i) the extent to which the cluster hypothesis holds, as measured by a specific test, and, (ii) the performance of cluster-based retrieval methods. Our main findings are that the cluster hypothesis can hold on the Web as it holds for newswire corpora; and, that using cluster-based retrieval methods can yield much merit for both settings.

Acknowledgments We thank the reviewers for their comments. This work has been supported by and carried out at the Technion-Microsoft Electronic Commerce Research Center. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

5. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMASS at TREC 2004 — novelty and hard. In *Proceedings of TREC-13*, 2004.

[2] G. V. Cormack, M. D. Smucker, and C. L. A. Clarke. Efficient and effective spam filtering and re-ranking for large web datasets. *CoRR*, abs/1004.5168, 2010.

[3] W. B. Croft. A model of cluster searching based on classification. *Information Systems*, 5:189–195, 1980.

[4] A. El-Hamdouchi and P. Willett. Techniques for the measurement of clustering tendency in document retrieval systems. *Journal of Information Science*, 13:361–365, 1987.

[5] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR*, pages 76–84, 1996.

[6] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240, 1971.

[7] O. Kurland. The opposite of smoothing: A language model approach to ranking query-specific document clusters. In *Proceedings of SIGIR*, pages 171–178, 2008.

[8] O. Kurland. Re-ranking search results using language models of query-specific clusters. *Journal of Information Retrieval*, 12(4):437–460, August 2009.

[9] O. Kurland and L. Lee. Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proceedings of SIGIR*, pages 83–90, 2006.

[10] O. Kurland and L. Lee. Clusters, language models, and ad hoc information retrieval. *ACM Transactions on information systems*, 27(3), 2009.

[11] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of SIGIR*, pages 120–127, 2001.

[12] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of SIGIR*, pages 186–193, 2004.

[13] X. Liu and W. B. Croft. Experiments on retrieval of optimal clusters. Technical Report IR-478, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts, 2006.

[14] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *Proceedings of ECIR*, pages 454–462, 2008.

[15] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *Proceedings of SIGIR*, pages 472–479, 2005.

[16] S.-H. Na, I.-S. Kang, and J.-H. Lee. Revisit of nearest neighbor test for direct evaluation of inter-document similarities. In *Proceedings of ECIR*, pages 674–678, 2008.

[17] J. Seo and W. B. Croft. Geometric representations for multiple documents. In *Proceedings of SIGIR*, pages 251–258, 2010.

[18] M. D. Smucker and J. Allan. A new measure of the cluster hypothesis. In *Proceedings of ICTIR*, pages 281–288, 2009.

[19] A. Tombros, R. Villa, and C. van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management*, 38(4):559–582, 2002.

[20] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

[21] E. M. Voorhees. The cluster hypothesis revisited. In *Proceedings of SIGIR*, pages 188–196, 1985.

[22] P. Willett. Query specific automatic document classification. *International Forum on Information and Documentation*, 10(2):28–32, 1985.

[23] L. Yang, D. Ji, G. Zhou, Y. Nie, and G. Xiao. Document re-ranking using cluster validation and label propagation. In *Proceedings of CIKM*, pages 690–697, 2006.

[24] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR*, pages 334–342, 2001.