AUTOCORRELATION AND REGULARIZATION OF QUERY-BASED INFORMATION RETRIEVAL SCORES

A Dissertation Presented

by

FERNANDO DIAZ

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2008

Computer Science

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
FEB 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Autocorrelation and Regularization of Query-Based Information Retrieval Scores	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Massachusetts Amherst,Department of Computer Science,Amherst,MA,01003**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**145**	

AUTOCORRELATION AND REGULARIZATION OF QUERY-BASED INFORMATION RETRIEVAL SCORES

A Dissertation Presented

by

FERNANDO DIAZ

Approved as to style and content by:

James Allan, Chair

W. Bruce Croft, Member

Sridhar Mahadevan, Member

John Staudenmayer, Member

Andrew Barto, Department Chair
Computer Science

a mis padres

ACKNOWLEDGMENTS

Some time ago in Peru, my father, Luis Diaz, a proud native of Cascas, met my mother, Dora Bieli-Bianchi, an equally-proud native of Moche. They fell in love, married, and in 1970 moved to the United States, initially as a temporary stay. Many factors made this stay permanent. The arrival of my two brothers and me almost certainly played an important part. Because of their great sacrifices, amazing inspiration, and constant support, I dedicate this dissertation to my parents. Not a single page would exist without them.

I would like thank a small crowd of individuals who indirectly pushed me into information retrieval. Dan Hepp at Proquest for introducing me to statistical natural language processing, Professor Christopher Achen for introducing me to formal modeling, and Rick McDermot for allowing his password to be compromised so that I could spend much time in high school playing with Lexis-Nexis.

I began working with James Allan at the start of my second year. I am grateful for his patience with my distraction, false leads, and stubbornness. James gave me a lot of freedom both to fall down and to find my own voice. At the same time, he always pushed me for answers where I had only speculation.

Bruce Croft significantly influenced my approach to information retrieval. He introduced me to the mathematical and theoretical treatments of the field, as well as the experimental rigor required, and the countless gems hidden in many undigitized papers. I have attempted to keep these three things in mind throughout the creation of this thesis.

Sridhar Mahadevan introduced me to many of the technical approaches used in this thesis. In particular, I was fortunate enough to have participated in several seminars on spectral graph theory led by Sridhar. The influence of these seminars and Sridhar's enthusiasm can be felt in the final work.

I would also like to acknowledge the role played by Rosie Jones. In 2003, I interned with Rosie at Overture Research (now Yahoo! Research Labs). Rosie's excitement, encouragement, and phenomenal mentoring played crucial roles in my academic career. Our collaborations were always very fruitful and I value our continued communication.

Additionally, I would like to thank my fellow students at the University of Massachusetts. Specifically, I would like to thank Desislava Petkova, Hema Raghavan, and Yun Zhou for assistance with data and feedback on papers and ideas. Although my collaborations with Donald Metzler are unfortunately not contained in this thesis, his feedback on some of the analysis in this work was very helpful. In addition, I'd like to thank Trevor Strohman for his comments, technical assistance, and aesthetic advice. Andre Gauthier developed an amazing research infrastructure which made all of the experiments in this thesis tractable. Victor Lavrenko, although he only commented on my thesis work a few times, provided valuable insights into the relationship between regularization and pseudo-relevance feedback.

Outside of school, I received support from the Martin Allen, Michael O'Neill, Louis Theran, and Greg Druck. My time in the Pioneer Valley would have been cut short were it

not for the distractions provided by the Flywheel Collective, the Hampshire Creative Music Collective, the Ecstatic Yod Collective, the Autonomous Battleship Collective, the Schoolhouse (RIP), Ruckus Entertainment, Shinola Entertainment, and Eremite Records.

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

ABSTRACT

AUTOCORRELATION AND REGULARIZATION OF QUERY-BASED INFORMATION RETRIEVAL SCORES

FEBRUARY 2008

FERNANDO DIAZ

B.Sc., UNIVERSITY OF MICHIGAN ANN ARBOR

B.A., UNIVERSITY OF MICHIGAN ANN ARBOR

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

Query-based information retrieval refers to the process of scoring documents given a short natural language query. Query-based information retrieval systems have been developed to support searching diverse collections such as the world wide web, personal email archives, news corpora, and legal collections. This thesis is motivated by one of the tenets of information retrieval: the cluster hypothesis. We define a design principle based on the cluster hypothesis which states that retrieval scores should be locally consistent. We refer to this design principle as *score autocorrelation*. Our experiments show that the degree to which retrieval scores satisfy this design principle correlates positively with system performance. We use this result to define a general, black box method for improving the local consistency of a set of retrieval scores. We refer to this process as *local score regularization*. We demonstrate that regularization consistently and significantly improves retrieval performance for a wide variety of baseline algorithms. Regularization is closely related to classic techniques such as pseudo-relevance feedback and cluster-based retrieval. We demonstrate that the effectiveness of these techniques may be explained by their regularizing behavior. We argue that regularization should be adopted either as a generic post-processing step or as a fundamental design principle for retrieval models.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

In information retrieval, we develop systems to help a searcher locate relevant data hidden in some large set of retrievable items. Searchers will have diverse backgrounds and needs. A searcher may know a lot about the relevant data or very little; a searcher may demand a few relevant items or an exhaustive list of relevant items. Unfortunately, the size of the relevant data set is much, much smaller than the number of retrievable items. In fact, the relevant data set may consist of a single document, a small set of sentences, or even a one-word answer. If the collection is not text, the set of retrievable items may contain artifacts such as images or movies. Complicating matters, information retrieval systems need to perform the classification of documents into relevant and non-relevant sets for many, arbitrary queries.

The study of automatic information retrieval, beginning in the 1950s, has had a very long history in computer science (even longer if we include study outside of computer science). Over this period, scientists have developed a set of design principles for information retrieval systems. When presented with a new retrieval scenario, we design systems or models with these principles in mind. Several classic design principles include preferring documents with multiple query term matches to those with fewer, weighting terms according to their inverse document frequency, rewarding documents with query terms in close proximity, and favoring popular documents. We can also view these principles as heuristics which systems or models must satisfy in order to perform well [Fang et al., 2004].

One principle which lies at the foundation of information retrieval is the idea of document clustering. Originally proposed by Jardine and van Rijsbergen in 1971, the *cluster hypothesis* can be stated as [van Rijsbergen, 1979],

> Closely associated documents tend to be relevant to the same requests.

Jardine and van Rijsbergen were interested in measuring the degree to which, for a given request or query, associated documents tended to has the same relevance to the searcher. In this thesis, we study topic-based information retrieval. Therefore, when we refer to a document as relevant, we mean that the document satisfies the query's topical requirement. This definition is commonly used in the information retrieval community, most visibly in the TREC conferences. Similarly, when we refer to the associations between documents, we mean the topical associations between documents. Topical association can be measured in different ways. For example, Jardine and van Rijsbergen investigate topical associations and use metrics such as term overlap statistics and the cosine correlation in order to detect topical associations.

The cluster hypothesis is one of the tenets of information retrieval, oft-cited and the motivation for numerous algorithms. Examples include clustering the corpus and retrieving a single cluster, locally interpolating document representations, and reducing the dimensionality of the entire semantic space. Unfortunately, the algorithms motivated by the cluster hypothesis are quite varied and it is often difficult to measure how exactly the design principle is being incorporated into a system.

In the first part of this thesis, we will focus on testing the cluster hypothesis. Consider the small set of documents in Figure 1.1 and their relationship to the query "dog". We observe that document (a) is clearly relevant because it discusses "dogs" explicitly and frequently; document (b) is much less relevant because it mostly discusses cats and only refers to "dogs" in passing; documents (c) and (d) are also relevant even though they use the scientific term "canine" instead of "dog". There is evidence for the cluster hypothesis if, given the query "dog", the relevance of document (a) implies the relevance of documents (c) and (d). We will develop a test to measure the extent to which, for a scoring of documents, the cluster hypothesis is satisfied. This measure, which we refer to as *score autocorrelation*, detects the degree to which a system scores documents consistently. Assume that, given the query "dog", we request that a retrieval system score the documents in Figure 1.1. A retrieval based solely on term frequencies may produce a set of scores $[3, 1, 0, 0]$ for documents (a), (b), (c), and (d). This retrieval would receive a low score autocorrelation because the scores of related documents (a), (c), and (d) are very different. A second retrieval system may produce the set of scores $[3, 0, 2.5, 2.5]$. This retrieval would receive a higher autocorrelation because scores of related documents are more consistent.

We adopt an autocorrelation measure from spatial data analysis referred to as the Moran autocorrelation. We will show that the autocorrelation of a set of retrieval scores in an affinity space induced by a similarity measure accurately captures the behavior in our example. With this consistency measure in hand, we will conduct a series of descriptive experiments measuring the correlation between consistency and system performance. The experiments in the first part of this thesis demonstrate the following two results,

1. The local consistency of a retrieval correlates positively with performance.

2. Many retrieval models fail to produce autocorrelated scores.

We demonstrate these results for a large set of baseline retrieval models over a diverse set of retrieval scenarios.

In the second part of this thesis, we propose a method for improving the effectiveness systems which fail to produce autocorrelated scores by improving local score consistency. Beginning with the scores from some baseline retrieval method, we develop an optimization problem to find a new set of scores which maximizes the local consistency. We refer to the process of finding more consistent scores as *local score regularization*. The intuition behind our solution is simple. Recall our documents in Figure 1.1. If the initial retrieval produces the scores $[3, 1, 0, 0]$, we will search the space of all score vectors for one which improves the consistency between documents (a), (c), and (d). The output of our system will be this score vector.

We adopt a regularization method based on the graph Laplacian. We will show that using the Laplacian directly models the introduction of local consistency into a set of retrieval

The **dog** (*Canis lupus familiaris*) is a domestic subspecies of the wolf, a mammal of the Canidae family of the order Carnivora. The term encompasses both feral and pet varieties and is also sometimes used to describe wild canids of other subspecies or species. The domestic **dog** has been (and continues to be) one of the most widely-kept working and companion animals in human history, as well as being a food source in some cultures. The **dog** is also the first animal from Earth to enter into space and fly into orbit.

(a)

The Cat (*Felis silvestris catus*), also known as the Domestic Cat or House Cat to distinguish it from other felines, is a small carnivorous species of mammal that is often valued by humans for its companionship and its ability to hunt vermin. It has been associated with humans for at least 9,500 years.

Cats, like **dogs**, are digitigrades: they walk directly on their toes, the bones of their feet making up the lower part of the visible leg.

(b)

Molecular systematics indicate that the domestic canine (*Canis lupus familiaris*) descends from one or more populations of wild wolves (*Canis lupus*). As reflected in the nomenclature, canines are descended from the wolf and are able to interbreed with wolves.

(c)

Canis lupus familiaris

(d)

Figure 1.1. Four documents related to the query "dog". All content appropriated from Wikipedia.

scores. With this algorithm in hand, we will conduct a series of effectiveness experiments measuring the improvement in performance garnered by the introduction of local consistency. The experiments in the second part of this thesis demonstrate the following two results,

1. Improving the local consistency of a system improves performance.
2. Many performance-improving retrieval methods can be seen as indirectly improving consistency

Again, we perform our experiments across a diverse set of tasks, demonstrating clear benefits to applying regularization. We will describe in detail our performance improvements and the tasks for which regularization is appropriate. Specifically, we will argue that regularization is a method well-suited for high-recall tasks which require inspecting deep into the ranked list. Because one of our fundamental data structures is the similarity matrix between documents, we will analyze the numerical stability of regularization as a function of changes to this similarity matrix. Furthermore, the relationship of regularization to the cluster hypothesis allows us to directly analyze classic information retrieval approaches from the perspective of regularization and allows us to develop new regularization-based methods for novel tasks.

Although many of the approaches we use in this thesis are new to information retrieval, the arguments and motivations are classic. This thesis makes the following contributions to information retrieval,

1. **A precision prediction method directly derived from the Voorhees cluster hypothesis test.** We develop a new precision prediction method directly related to the Voorhees cluster hypothesis test. This measure is attractive because of its relationship to established tests in spatial data analysis.
2. **A large-scale analysis of the local score consistency in retrieval systems.** We measure the amount of local score consistency for a large population of retrieval submissions to various TREC competitions. The results indicate that many retrieval systems do not consider local score consistency.
3. **A consistently beneficial document re-ranking algorithm.** We describe a new method based on the graph Laplacian for re-ranking documents based on improving local score consistency. We demonstrate that this algorithm is generally applicable and easily-extendable into new domains.
4. **A regularization-based perspective on pseudo-relevance feedback.** We present an extended discussion of the relationship between regularization and previous research, concluding that some of the success of these methods may be explained by their effect on local score consistency.

The remainder of this dissertation proceeds as follows,

Chapter 2: Preliminaries We define the information retrieval task and review several classic and modern algorithms for retrieval. In addition, we survey text similarity measures used in information retrieval.

Chapter 3: The Cluster Hypothesis in Information Retrieval We describe the Jardine-van Rijsbergen and Voorhees tests of the cluster hypothesis. We present theoretical and experimental arguments for using the Voorhees measure.

Chapter 4: Autocorrelation of Retrieval Scores Starting from the Voorhees test, we develop score autocorrelation as measure of local score consistency. We test the correlation between local consistency and performance for a large set of retrieval methods.

Chapter 5: Regularization of Retrieval Scores We describe an algorithm for improving the local score consistency of arbitrary retrieval methods. We provide experimental evidence demonstrating the effectiveness of regularization.

Chapter 6: Relationship to Other Retrieval Methods We present a comprehensive and detailed comparison of regularization to a previous retrieval methods.

Chapter 7: Stability of Regularization We analyze the numerical stability of regularization subject to different similarity measures.

Chapter 8: Extensions and Future Work We describe extensions of the regularization for relevance feedback, cross-lingual retrieval, optimal set retrieval, and cross-media retrieval.

CHAPTER 2

PRELIMINARIES

In this chapter, we will review several retrieval models referenced in this thesis. This survey is by no means exhaustive with respect to models or techniques within each model. Instead, we focus on models representative of classic and state of the art approaches. We will define models in their standard notation as well as matrix notation. Matrix and vector conventions are presented in Table B.1. This chapter provides a background for algorithms and techniques used later in this thesis.

2.1 The Document Collection

A *collection* is a set of n documents which exist in an m-dimensional vector space where m is the size of the vocabulary and elements of the vectors represent the frequency of the term in the document. The set of documents in the collection will often be indicated as $\mathcal{D} = \{i | 1 \leq i \leq n\}$. We define for each document $i \in \mathcal{D}$ a column vector, $\mathbf{d}_i$, where each element of the vector represents the frequency of the term in document i; we refer to this as the *document vector*. Transposing and stacking up the n document vectors defines the $n \times m$ collection matrix $\mathbf{C}$.

We define other symbols in Table B.2. Elaborations of definitions will occur when notation is introduced.

2.2 Retrieval Scores

A *set retrieval model* assigns a binary prediction of relevance to each document in the collection. The user then scans those documents predicted to be relevant. We can see this as a mapping or *function* from documents in the collection to a binary value. Mathematically, given a query, q, a set retrieval model provides a function, $f_q : \mathcal{D} \rightarrow \{0, 1\}$, from documents to labels; we refer to f_q as the *initial score function* or initial retrieval for a particular query. The argument of this function is the retrieval system's representation of a document. The values of the function provide the system's labeling of the documents. Notice that we index functions by the query. We note this to emphasize the fact that, in information retrieval, the score function over all documents will be different for each query. Although we drop the index for notational convenience, this function is always associated with a particular query.

A *ranked retrieval model* assigns some rank or score to each document in the collection and ranks documents according to the score. The user then scans the documents according to the ranking. The score function for a ranked retrieval model maps documents to real values.

Given a query, q, the model provides a function, $f_q : \mathcal{D} \rightarrow \Re$, from documents to scores. The values of the function provide the desired ranking of the documents.

In this section, we will review several classic and state-of-the-art ranked retrieval models. Each retrieval model provides a method for defining a function $f_q : \mathcal{D} \rightarrow \Re$. Since each function can be treated as a set of scores assigned to an indexed set of documents, we represent score function using the vector $\mathbf{y} \in \Re^n$.

2.2.1 Vector Space Model

The vector space model is one of the most general information retrieval models [Salton, 1968]. By treating a query as a very short document, documents and queries can be represented in a shared, m-dimensional space and scores can be computed using the cosine similarity measure.

In Section 2.1, we described document vectors as consisting of raw term frequencies. In practice, the elements of these vectors (and therefore $\mathbf{C}$) are adjusted to weight terms according to their relative importance in the document and discriminativeness in the collection. These are referred to as the term frequency or tf weight and inverse document frequency or idf weight, respectively. Using the BM25 weights [Robertson and Walker, 1994], documents are represented as,

$$\tilde{d}_i = \underbrace{\frac{d_i(k+1)}{d_i + k\left((1-b) + b\left(\frac{l_i}{||\mathbf{l}||_1/n}\right)\right)}}_{Okapi\ term\ frequency} \times \underbrace{\log\left(\frac{(n+0.5) - c_i}{0.5 + c_i}\right)}_{inverse\ document\ frequency} \tag{2.1}$$

where $\mathbf{d}$ is a length-m document vector where elements contain the raw term frequency, the vector $\mathbf{l}$ is the length-n vector of document lengths, $l_i = ||\mathbf{d}_i||_1$, and $\mathbf{c}$ is the length-m document frequency vector.

The cosine similarity between the query and document can be computed as,

$$cos(\mathbf{q}, \tilde{\mathbf{d}}) = \frac{\mathbf{q}^\top \tilde{\mathbf{d}}}{\|\mathbf{q}\|_2 \times \|\tilde{\mathbf{d}}\|_2} \tag{2.2}$$

which is equivalent to the inner product between L_2-normalized vectors. When discussing the vector space model, we will assume that the rows of $\mathbf{C}$ are reweighted according to Equation 2.1 and L_2-normalized. We will also assume that the query vector, $\mathbf{q}$, is L_2 normalized. Using this notation, the scores for all documents in the collection can be represented as,

$$\mathbf{y} = \mathbf{C}\mathbf{q} \tag{2.3}$$

Pseudo-relevance feedback or *query expansion* refers to the technique of using information from the top r documents retrieved by the original query. The system then performs a second retrieval using combination of this information and the original query. One way to incorporate this information is to assume that the top r documents are relevant [Croft and Harper, 1979]. If the top r documents are assumed to be relevant, we can use the classic Rocchio technique for incorporating additional terms [Rocchio, 1971]. Let the *pseudo-relevant* set be

R and $r = |R|$. In Rocchio feedback, we linearly combine the vectors of documents in R with the original query vector, $\mathbf{q}$. The modified query, $\tilde{\mathbf{q}}$, is defined as,

$$\tilde{\mathbf{q}} = \mathbf{q} + \frac{\alpha}{r} \sum_{j \in R} \mathbf{d}_j \tag{2.4}$$

where α is the weight placed on the information from the pseudo-relevant documents. We can use this new representation to score documents by their similarity to $\tilde{\mathbf{q}}$,

$$\tilde{\mathbf{y}} = \mathbf{C}\tilde{\mathbf{q}} \tag{2.5}$$

2.2.2 Language Model Scores

In the language modeling approach to information retrieval, terms found in a document represent samples from some underlying m-dimensional multinomial over terms in the vocabulary [Croft and Lafferty, 2003]. Each document in the collection is associated with a unique multinomial which is referred to as the *document language model*. A document language model can be estimated from the terms occurring in the text. A user's query is treated as an unordered bag of words. Then, given a query, documents can be ranked by their probability of having generated the query sequence. The intuition behind this ranking is that documents which are more likely to have generated the query are more likely to be relevant. This ranking method is referred to as *query likelihood retrieval*.

There are many ways to estimate a document language model. Let $\mathbf{d}$ contain the raw term frequencies for a document and $P(w|\theta_d)$ be the estimation of the document language model. The *maximum likelihood estimate* of $P(w|\theta_d)$ defined as,

$$P(w|\theta_d) = \frac{d_i}{\|\mathbf{d}\|_1} \tag{2.6}$$

When estimating a distribution, especially with a small sample, it is statistically attractive to reserve some probability mass for unseen events. For example, given a document about dogs, even if we never saw the word "canine" or "cat", we would like to think that, if the author continued writing, these words would occur with some probability. The assignment of non-zero weights to unseen terms is referred to as *smoothing*. One popular and effective method for smoothing document models is to use the conjugate prior of the distribution; for the multinomial, this would be the Dirichlet distribution. Using Dirichlet smoothing, We estimate the document language model as,

$$\tilde{d}_i = \frac{d_i + \mu P(w_i|\theta_C)}{||\mathbf{d}||_1 + \mu} \tag{2.7}$$

where $P(w|\theta_C)$ is the maximum likelihood collection model defined as,

$$P(w_j|\theta) = \frac{\sum_i C_{ij}}{\sum_{ik} C_{ik}} \tag{2.8}$$

A comparison of alternative smoothing methods for information retrieval can be found in [Zhai and Lafferty, 2004].

Given a query, we rank documents according to their query likelihood. We can write the query as a length-m vector where elements contain the frequency of each term in the query. Using the common assumption that query terms are independently sampled from the underlying model, a document's score can be written as,

$$\begin{aligned} P(Q|\theta_d) &= \prod_{i=1}^{m} P(w_i|\theta_d)^{q_i} \\ &\stackrel{\text{rank}}{=} \sum_{i=1}^{m} q_i \log P(w_i|\theta_d) \\ &\propto \sum_{i=1}^{m} \frac{q_i}{\|\mathbf{q}\|_1} \log P(w_i|\theta_d) \\ &= \sum_{i=1}^{m} P(w_i|\theta_q) \log P(w_i|\theta_d) \end{aligned} \tag{2.9}$$

In the second line of this derivation, we take the logarithm of the product which preserves the rank ordering of documents and results in a linear scoring function. In the third line, we multiply by the reciprocal of the query length; which also preserves the ordering of the documents. Finally, we recognize the maximum likelihood query model, $P(w|\theta_q)$, in the formula using Equation 2.6. Note here that the score is actually the cross entropy between the document model, $P(w|\theta_d)$, and the query model, $P(w|\theta_q)$. We can write this function in vector notation as,

$$P(Q|\theta_d) \stackrel{\text{rank}}{=} [\log \tilde{\mathbf{d}}]^{\mathsf{T}} \mathbf{q} \tag{2.10}$$

where $\mathbf{q}$ and $\mathbf{d}$ are language models. When discussing the language model approaches, we will assume that the rows of $\mathbf{C}$ are smoothed according to Equation 2.7. We will also assume that the query vector, $\mathbf{q}$, is L_1 normalized. Using this notation, the scores for all documents in the collection can be represented as,

$$\mathbf{y} = (\log \mathbf{C})\, \mathbf{q} \tag{2.11}$$

In the language modeling framework, pseudo-relevance feedback can be defined in several ways. We focus on the "relevance model" technique [Lavrenko, 2004]. In relevance modeling, the original scores are used as weights for the estimated relevance model. The relevance model, $P(w|\theta_R)$, is formally constructed by interpolating the maximum likelihood query model, $P(w|\theta_q)$, and relevance-weighted document models, $P(w|\theta_d)$,

$$P(w|\theta_R) \;=\; \lambda P(w|\theta_q) + (1-\lambda) \left(\sum_{d \in R} \frac{P(Q|\theta_d)}{\mathcal{Z}} P(w|\theta_d) \right) \tag{2.12}$$

where $\mathcal{Z} = \sum_{D \in R} P(Q|\theta_d)$ which means we are using an L_1 normalized version of $\mathbf{y}$. This is clearer if we represent $P(w|\theta_R)$ in matrix notation,

$$\tilde{\mathbf{q}} \;=\; \lambda \mathbf{q} + \frac{(1-\lambda)}{||\mathbf{y}||_1} \mathbf{C}^{\mathsf{T}} \mathbf{y} \tag{2.13}$$

Cross entropy scoring can be used because $\tilde{\mathbf{q}}$ is a language model. The document scores after pseudo-relevance feedback are,

$$\tilde{\mathbf{y}} = (\log \mathbf{C})\, \tilde{\mathbf{q}} \tag{2.14}$$

2.2.3 Feature-Based Retrieval

Both the vector space model and the unigram language model approaches to information retrieval represent documents as unordered bags of words. In feature-based retrieval, documents are still represented as vectors. However, the components of these vectors do not represent words. Instead, components represent the features we expect relevant documents to have. We expect relevant documents to contain many occurrences of query terms which may be represented as the feature. Given the query sequence Q and a document D, this feature may be defined by

$$\psi_t(Q, D) = \sum_i q_i d_i \tag{2.15}$$

where $\mathbf{q}$ is a vector of query terms, $\mathbf{d}$ is our $m \times 1$ document term vector, and ψ is our document feature vector. Alternatively, we could use one of the term-based scores computed by bag of words approaches. The attractive aspect of feature-based retrieval is that we can represent more complex features in ψ as well. For example, if we are interested in the proximity between query terms, we might define the following feature,

$$\psi_o(Q, D) = \sum_{ij} q_i q_j \tilde{d}_{ij} \tag{2.16}$$

where $\tilde{\mathbf{d}}$ is a $m^2 \times 1$ document proximity vector indicating the frequency of co-occurrence of i and j within some window of terms. We can also define query-independent features such as the PageRank or document quality [Brin and Page, 1998; Zhou and Croft, 2005].

In this thesis, Metzler's Markov random field (MRF) model of retrieval represents the family of feature-based retrieval methods [Metzler and Croft, 2005]. The MRF model of retrieval computes a document the joint probability of D and Q as

$$P_{G,\Lambda}(Q, D) = \frac{\prod_i \psi_i(Q, D)}{Z_\Lambda} \tag{2.17}$$

where ψ is a feature vector and $Z_\Lambda = \sum_{D,Q} \prod_i \psi_i$. If we take the logarithm of this equation, we can derive the following ranking function,

$$\log P_{G,\Lambda}(Q, D) \stackrel{\text{rank}}{=} \underbrace{\lambda_{T_D} \sum_{c \in T_D} f_{T_D}(c)}_{\text{terms}} + \underbrace{\lambda_{O_D} \sum_{c \in O_D} f_{O_D}(c)}_{\text{ordered pairs}} + \underbrace{\lambda_{U_D} \sum_{c \in U_D} f_{U_D}(c)}_{\text{unordered pairs}}$$

where T_D are terms in Q, O_D are ordered pairs of terms in Q, and U_D are unordered pairs of terms in Q. The operators f_* are functions of the occurrence of those terms (or pairs) in D.[1]

[1] The intimidating math often used to describe the MRF betrays the simple reduction to a linear combination of document scores. Similar methods have previously been used in metasearch [Montague and Aslam, 2001]. In fact, the parallels between feature-based methods and metasearch have allowed feature-based methods to be applied directly to the metasearch problem [Carterette and Petkova, 2006; Yue et al., 2007].

2.2.4 Summary

There are a few observations about these retrieval functions worth noticing. First, in most cases, document scores are calculated independently. There is often no explicit representation of the score dependencies between documents. Second, we rank documents by decreasing scores. We are confident that the highly ranked documents are likely to relevant; we are less confident that the lower ranked documents are relevant. Treating a score as a confidence is different from treating it as a label. If we treat the score as a confidence of relevance, there is no way to represent confidence that a document not relevant. We will return to this thought near the end of the thesis. Finally, these functions in practice have very skewed distributions of scores; most of the n documents in the corpus and even most of the top $\tilde{n}$ retrieved documents have very low scores when compared to the top-ranked documents. We present the score distributions for a few queries in Figure 2.1. This behavior is consistent across most retrieval algorithms [Manmatha et al., 2001].

These retrieval models represent a very small but representative view of information retrieval methods. One good catalog of alternative retrieval methods can be found in proceedings of the Text Retrieval Conference (TREC) [Voorhees and Harman, 2001]. In order to demonstrate the generalizability of our results, we have attempted, when possible, to present experiments which use the algorithms presented in Sections 2.2.1-2.2.3 as well as the larger population of algorithms produced at the TREC conferences.

2.3 Inter-document Relationships

In many collections, documents exist as independent of one another; that is, there is no explicit relationship between any pair of documents. A news corpus, a collection of newswire articles, is known to have this property in many situations. However, we know that, while not explicit, relationships between documents indeed exist. For example, two news articles about "hostage-taking" have shared topic and therefore have a topical relationship. In this section, we review prior work which studied inter-document relationships. We will describe the definition and detection of inter-document relationships.

Depending on one's perspective, two documents may be related in different ways. For example, documents may be related by a citation, a hyperlink, coauthors, or shared topics. When studying some property of the collection, we often must select from the set of all possible relationships. The task of interest should guide the selection an appropriate inter-document relationship (or set of relationships). For example, because this thesis is concerned with retrieving documents on a certain topic, we focus on inter-document topical relationships. Although other relationships might—and often do—correlate with shared topics, the fundamental task is driven by modeling the topics discussed in documents.

Inter-document relationships traditionally determined by explicit labeling can also be inferred from similarity of the language shared between two documents. Several similarity measures have been proposed and are the foundation of many classic clustering algorithms [Lance and Williams, 1967]. A similarity measure's effectiveness can be determined indirectly by its influence on a task such as document classification or query-based retrieval. In the topic detection and tracking (TDT) literature, inter-document similarity is evaluated directly by comparing system predictions with human judgments; because the TDT program

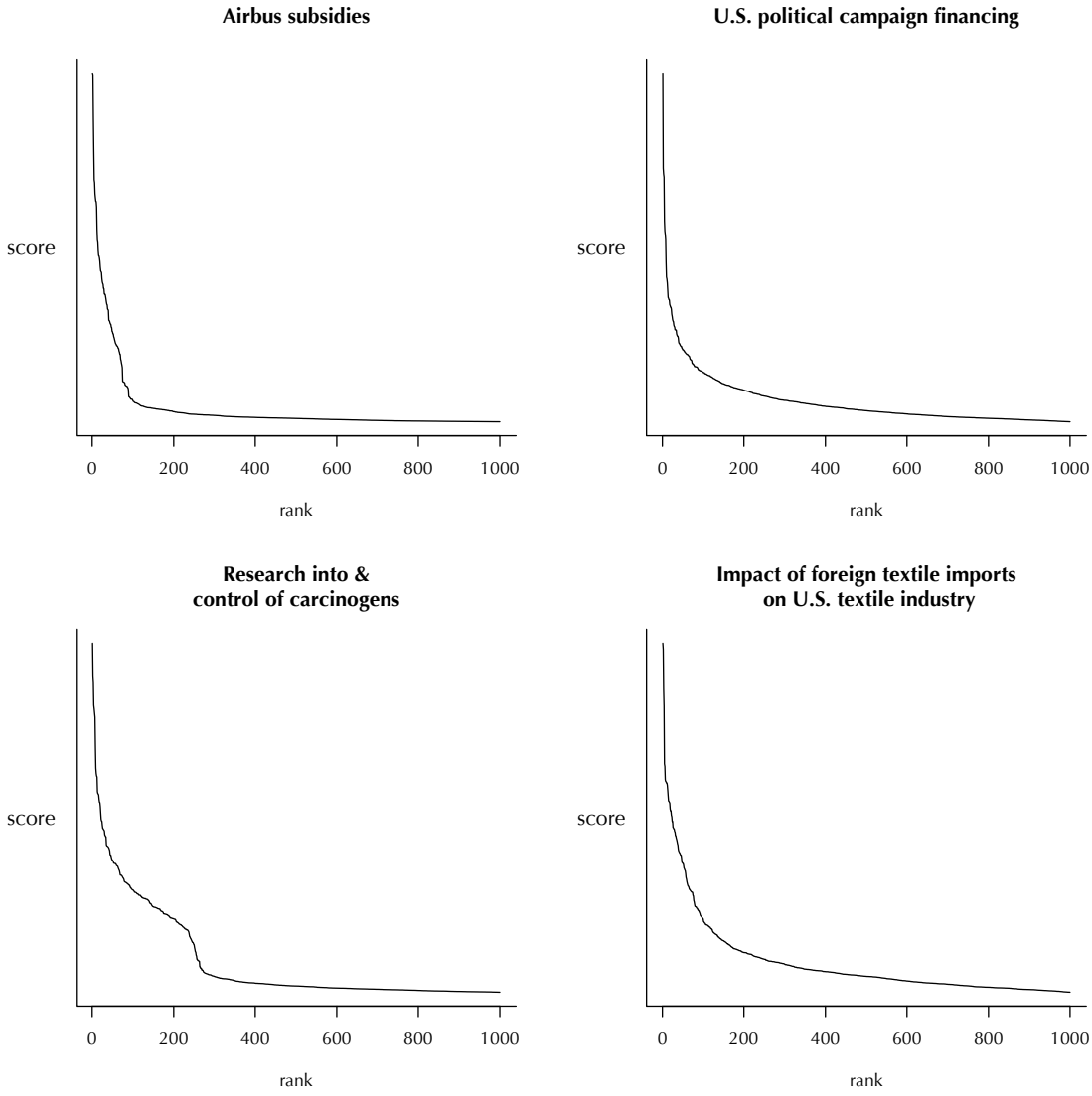


Figure 2.1. Score distributions for several queries from the trec12 query set. Query scores were computed for the top 1000 documents retrieved using query likelihood retrieval.

was conducted in the context of news documents, this evaluation is referred to as "story link detection" [Allan, 2002].

In this thesis, we will focus on two high-performing approaches to link detection: the classic vector space model and language model [Chen et al., 2004]. These two approaches are related to the score functions described in Section 2.2. Both of these approaches represent documents using bags of words, ignoring proximity or phrase information. Our focus on unigram techniques is motivated by the lack of significant improvement when dependencies are considered [Bekkerman and Allan, 2004; Nallapati and Allan, 2002].

Pairwise relationships between documents will be represented by the $n \times n$ symmetric matrix, $\mathbf{A}$.[2] Each similarity measure will define all of the entries for $\mathbf{A}$.

2.3.1 Cosine Similarity

Recall that that in the vector space model, we assume that each document vector, $\mathbf{d}_i$, is weighted by tf.idf and L_2-normalized. The cosine between document vectors determines affinity,

$$\begin{aligned} \cos(\mathbf{d}_i, \mathbf{d}_j) &= \langle \mathbf{d}_i, \mathbf{d}_j \rangle \\ &= \mathbf{d}_i^{\mathsf{T}} \mathbf{d}_j \end{aligned} \tag{2.18}$$

The affinity matrix is defined by,

$$\mathbf{A}_{cos} = \mathbf{C}\mathbf{C}^{\mathsf{T}} \tag{2.19}$$

2.3.2 Language Model Similarity

When represented as language models, documents can be compared using a multinomial similarity measure.

The Kullback-Leibler divergence between two distributions is a well-known, theoretically-motivated measure of dissimilarity.

$$D_{KL}(\mathbf{d}_i \| \mathbf{d}_j) = H(\mathbf{d}_i) - \langle \mathbf{d}_i \log(\mathbf{d}_j) \rangle \tag{2.20}$$

where $H(\mathbf{d})$ is the information entropy defined by $H(\mathbf{d}) = \langle \mathbf{d}, \log(\mathbf{d}) \rangle$; the second term is the cross entropy between i and j. We should make a few observations about the Kullback-Leibler divergence. First, the measure is asymmetric (ie, $D_{KL}(\mathbf{d}_i \| \mathbf{d}_j) \neq D_{KL}(\mathbf{d}_j \| \mathbf{d}_i)$). Unfortunately, the semantics of the asymmetry are unclear. This makes adoption of the Kullback-Leibler divergence problematic. Second, although the measure is zero when two multinomial are equal, there is no theoretical maximum for arbitrary multinomials.

[2] We assume symmetric relationships representing the sharing of a topic. We exclude asymmetric topical relationships because they introduce assumptions such as containment and entailment which have not been throughly studied in the information retrieval literature.

In order to address some of the problems with the Kullback-Leibler divergence, Lin proposed an alternative measure inspired by Shannon entropy [Lin, 1991]. The Jensen-Shannon divergence is defined as,

$$JS_{\pi}(\mathbf{d}_i, \mathbf{d}_j) = H(\pi_i \mathbf{d}_i + \pi_j \mathbf{d}_j) - (\pi_i H(\mathbf{d}_i) + \pi_j H(\mathbf{d}_j)) \tag{2.21}$$

where $\pi_i + \pi_j = 1$. In order for this measure to be symmetric, we set $\pi_i = \pi_j$. This results in the following derivation,

$$H\left(\frac{1}{2}(\mathbf{d}_i + \mathbf{d}_j)\right) - \frac{1}{2}(H(\mathbf{d}_i) + H(\mathbf{d}_j)) \propto D_{KL}\left(\mathbf{d}_i \middle\| \frac{1}{2}(\mathbf{d}_i + \mathbf{d}_j)\right) + D_{KL}\left(\mathbf{d}_j \middle\| \frac{1}{2}(\mathbf{d}_i + \mathbf{d}_j)\right)$$

The Bhattacharyya distance measures the angle between multinomials and has been used for link detection in the past [Chen et al., 2004]. The Bhattacharyya distance is defined as,

$$\mathcal{B}(\mathbf{d}_i, \mathbf{d}_j) = \left\langle \sqrt{\mathbf{d}_i}, \sqrt{\mathbf{d}_j} \right\rangle \tag{2.22}$$

A related measure is the Hellinger distance, defined as,

$$\begin{aligned} \mathcal{H}(\mathbf{d}_i, \mathbf{d}_j) &= \left\| \sqrt{\mathbf{d}_i} - \sqrt{\mathbf{d}_j} \right\|_2^2 \\ &= 2(1 - \mathcal{B}(\mathbf{d}_i, \mathbf{d}_j)) \end{aligned} \tag{2.23}$$

The performance of these measures, although comparable, have not shown to improve performance for tasks such as link detection or clustering [Chen et al., 2004].

Lebanon and Lafferty propose a kernel for multinomials based on diffusion over the multinomial manifold [Lafferty and Lebanon, 2005]. This affinity measure between two distributions is motivated by the Fisher information metric and defined as,

$$\mathcal{K}_t(\mathbf{d}_i, \mathbf{d}_j) = \exp\left(-t^{-1} \arccos^2 \mathcal{B}(\mathbf{d}_i, \mathbf{d}_j)\right) \tag{2.24}$$

where t is a parameter controlling the decay of the affinity. The diffusion kernel has been shown to be a good affinity metric for tasks such as text classification [Lafferty and Lebanon, 2005]. In fact, when two documents are very similar, the diffusion kernel is nearly-equivalent to the square root of the Kullback-Leibler divergence. For text, the Bhattacharyya distance and the multinomial diffusion kernel are attractive for theoretical reasons. Lafferty and Lebanon note [Lafferty and Lebanon, 2005, p. 139],

> The Fisher information metric places greater emphasis on points near the boundary, which is expected to be important for text problems, which typically have sparse statistics.

For this reason, we adopt these measures in our experiments and define the following two similarity matrices,

$$\mathbf{A}_{\mathcal{B}} = \left(\sqrt{\mathbf{C}}\right)\left(\sqrt{\mathbf{C}}\right)^{\mathsf{T}} \tag{2.25}$$

$$\mathbf{A}_{\mathcal{K}_t} = \exp(-t^{-1} \arccos^2(\mathbf{A}_{\mathcal{B}})) \tag{2.26}$$

We plot the relationship between the Bhattacharyya distance and the diffusion kernel in Figure 2.2.

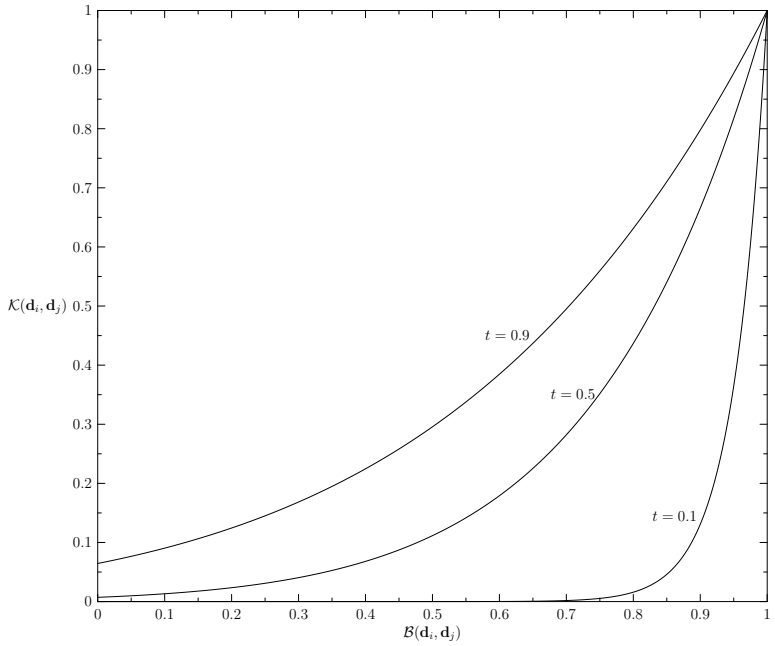


Figure 2.2. Relationship between Bhattacharyya and diffusion kernels.

2.3.3 Visualizing Inter-document Affinity

In this thesis, we will, at times, visualize the matrix **A** in order to support explanation. In this short section, we will describe the process for generating these two-dimensional visualizations. We caution that projecting from m dimensions to two obscures many potentially-interesting observations. Therefore, throughout this thesis, we will be using visual illustrations to provide intuition for algorithms and measurements, not evidence of effectiveness.

Assuming all documents share at least small subset of vocabulary, the affinity matrix, **A**, contains n^2 non-zero entries. We make the matrix sparser by including only the k-largest similarities for each document. More concretely, assume that $\mathcal{S}_i$ is the size k set of indexes of the maximum values in the ith row of **A**. The sparser matrix, **W**, is defined as,

$$W_{ij} = \begin{cases} A_{ij} & \text{if } i \in \mathcal{S}_j \text{ or } j \in \mathcal{S}_i \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

In addition to sparsifying **A**, we also make the affinity matrix tractable by using query-based samples of size $\tilde{n}$ from the collection. Because our analysis is query-based, this means we have a different matrix for each query.

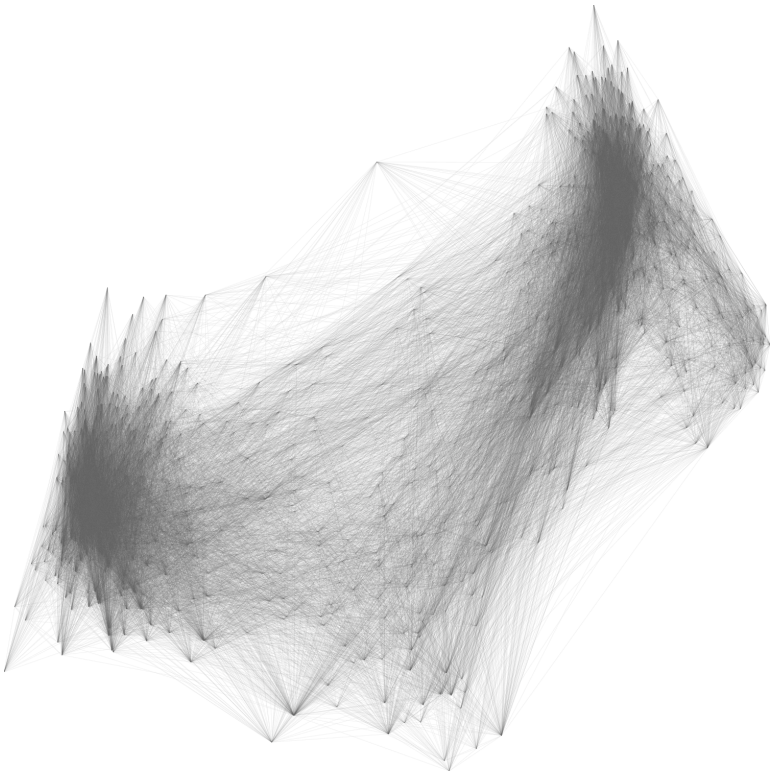
Given a pairwise affinity matrix, we can use a number of techniques for embedding the data in two dimensions. Later in this thesis, we will use the combinatorial Laplacian defined for **W** in order to analyze retrieval functions. The Laplacian provides a robust, diffusion-based embedding into two dimensions [Coifman and Lafon, 2006]. Alternatively, when considered a graph, the affinity matrix can be projected using spring-embedding or energy-based graph drawing techniques [Leuski, 2001; Adai et al., 2004]. We compare projections

based on the Laplacian and a spring-embedding in Figure 2.3. Although we notice that some of the coarse structure in the projections is similar, the embedding based on the Laplacian results in a less-intuitive rendering. Visualization based on the eigenvectors of the Laplacian arises from low frequency harmonics on the graph. This results in a nice visualization of coarse graph structure. Spring embedding captures lower level structure but introduces some high level error. Because retrieval focuses on fine grained clusters of documents, we adopt the spring embedding layout when illustrating data or algorithmic effects. Although we abandon the Laplacian for visualization, we will reintroduce it in Chapter 5 for analyzing score functions.

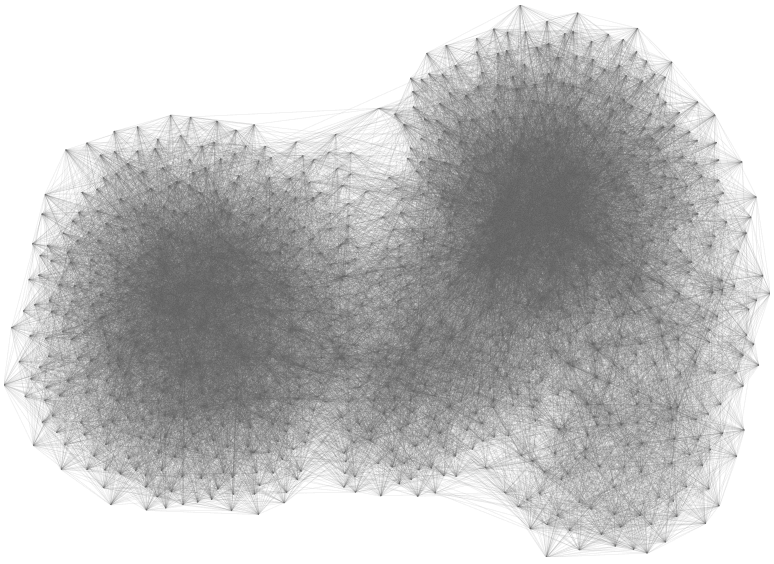
In this thesis, we use the Large Graph Layout spring-embedding algorithm to visualize document graphs [Adai et al., 2004]. We present an example graph for different values of k and $\tilde{n}$ in Figure 2.4.

2.3.4 Summary

Inter-document similarity, as presented here, reduces to a function of the inner product of two document vectors. As we mentioned earlier, approaches incorporating dependencies between dimensions have usually demonstrated only slight improvements in link detection. Our adoption of linear similarity measures will allow us to analyze a number of retrieval methods in a general way in Chapter 6.



(a) diffusion maps



(b) spring-embedding

Figure 2.3. Comparison of two-dimensional embedding using diffusion maps and Large Graph Layout.

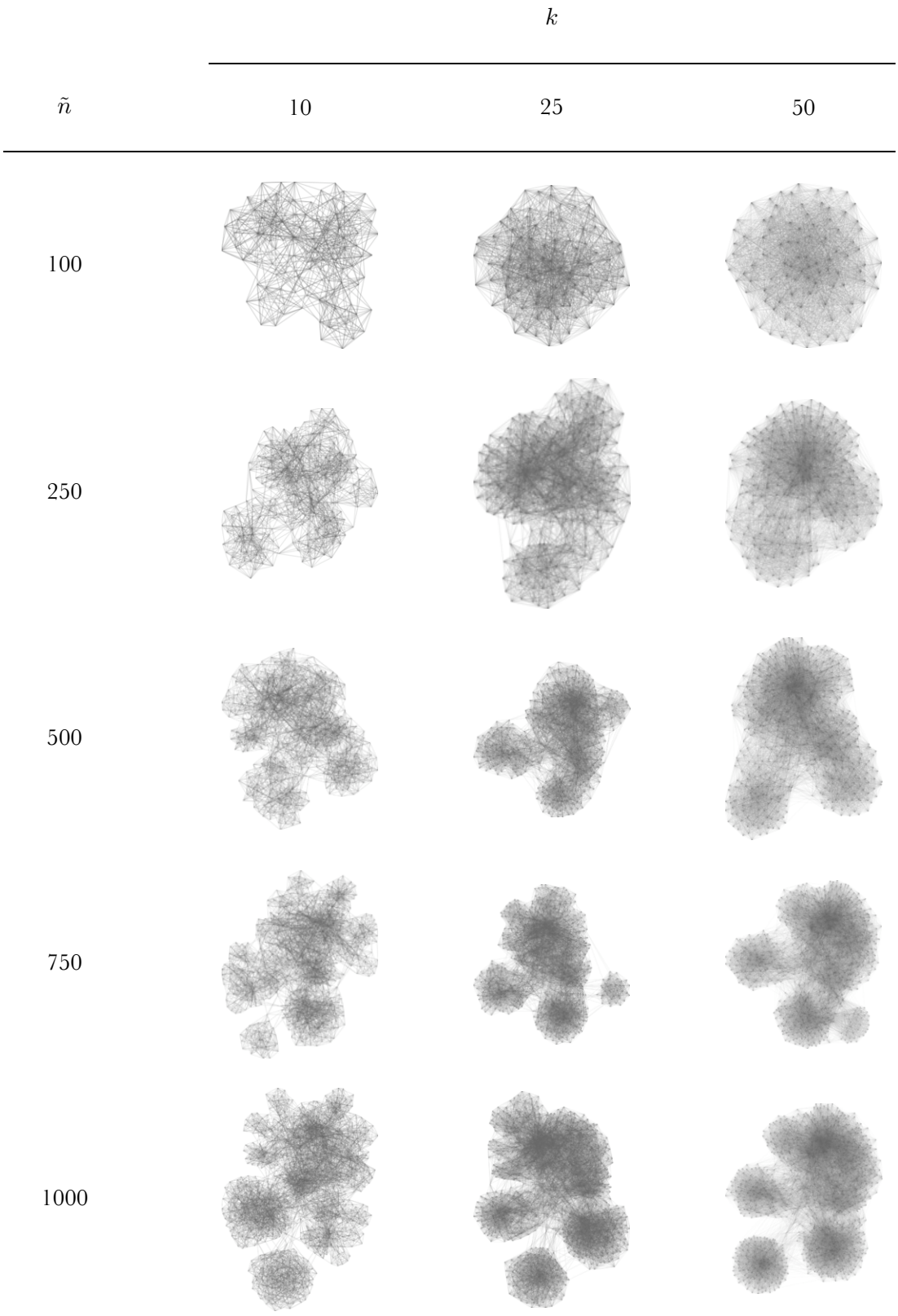


Figure 2.4. Document graphs for the TREC query "The Effectiveness of Medical Products and Related Programs Utilized in the Cessation of Smoking". Document graphs are built using the top $\tilde{n}$ retrieved documents. Edges are added for the k nearest neighbors of each document.

PART I

AUTOCORRELATION OF RETRIEVAL SCORES

CHAPTER 3

THE CLUSTER HYPOTHESIS IN INFORMATION RETRIEVAL

The cluster hypothesis, as posed by van Rijsbergen, states: "closely associated documents tend to be relevant to the same [queries]" [van Rijsbergen, 1979]. In general, this hypothesis has been tested using two approaches. The first approach, originally proposed by Jardine and van Rijsbergen, tests the degree to which relevant documents exist as a single, cohesive cluster distinct from the non-relevant documents. The second approach, advanced by Voorhees as an alternative to the Jardine-van Rijsbergen test, measures the degree to which relevant documents are related to other relevant documents. The confirmation of the cluster hypothesis for an individual query, justifies the incorporation of inter-document similarity into our final document ranking. Either test, Jardine-van Rijsbergen or Voorhees, motivates its own set of approaches. We will compare the assumptions and behaviors of the Jardine-van Rijsbergen and Voorhees tests, arguing that the Voorhees test is more robust and appropriate for information retrieval. This chapter provides a foundation for development of local score consistency in the next chapter.

3.1 The Jardine-van Rijsbergen Test

Jardine and van Rijsbergen test the cluster hypothesis by comparing the distribution of similarities between relevant documents and the distribution of similarities between relevant and non-relevant document; we refer to these distributions as RR and NR, respectively. In Figure 3.1, we indicate the submatrices of $\mathbf{W}$ representing the similarities between relevant and non-relevant documents. Histograms of similarities found in these submatrices let us estimate distributions of similarities within and between the classes. Jardine and van Rijsbergen test the following hypothesis,

Hypothesis 3.1. *The RR and NR distributions are well-separated.*

In Figure 3.2, we demonstrate with artificial data situations where the cluster hypothesis holds and when it does not. Jardine and van Rijbergen use visual inspection as evidence for and against the cluster hypothesis. In subsequent work, van Rijsbergen and Sparck Jones suggested that the degree to which collections satisfy the clustering hypothesis (by visual inspection) correlates strongly with retrieval performance [van Rijsbergen and Sparck Jones, 1973]. Quantitative comparisons between the two distributions might include comparing the means. Griffiths *et al* test for the cluster hypothesis by measuring the overlap between the two distributions in Figure 3.2 [Griffiths et al., 1986].

In the most general interpretation, Hypothesis 3.1 makes a conjecture about the relationship between *all* documents in the collection. That is, if we select two random documents

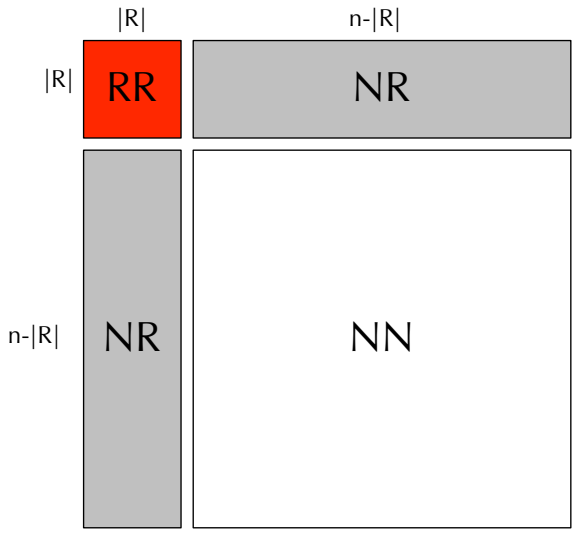


Figure 3.1. Matrix elements used for testing the cluster hypothesis.

from the collection and if they are topically related, then we should expect them to have the same relevance. In practice, authors often confine the analysis to the top $\tilde{n}$ documents retrieved from the query [Hearst and Pedersen, 1996].

3.1.1 Approaches Motived by the Jardine-van Rijsbergen Test

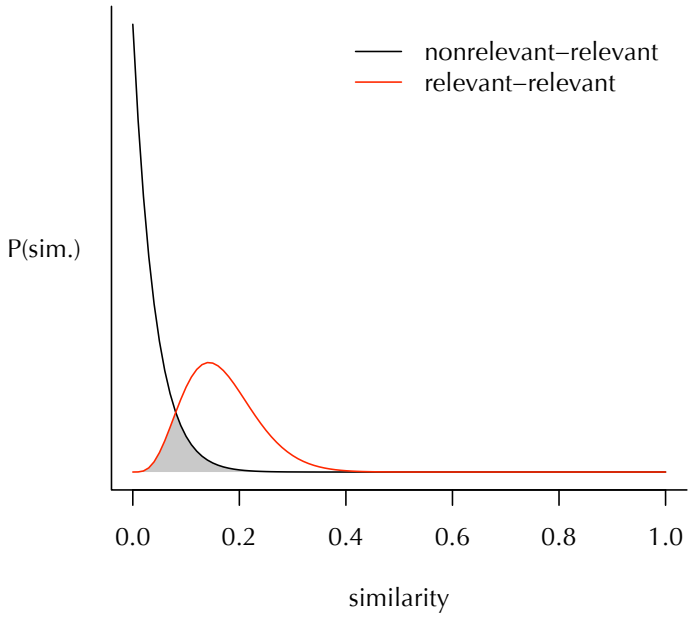
So far, we have described how one might test Hypothesis 3.1. If there is evidence that this hypothesis is true for a query or collection, how do we change our retrieval methods to exploit clustering? In the remainder of this section, we will review several techniques.

3.1.1.1 Clustering Documents

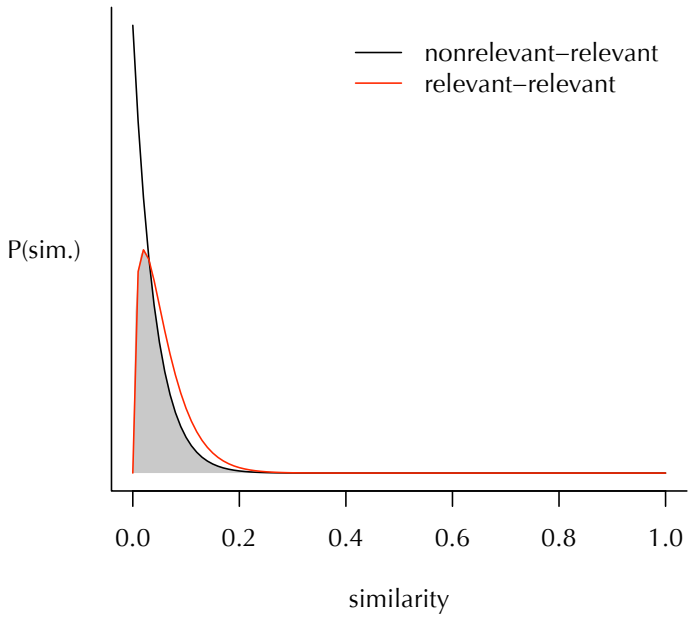
Clustering refers to the assignment of each document to one or more of k groups of documents. Documents within a cluster all share some topic. Clusters can be defined manually or automatically. Automatic clustering may refer to any number of algorithms and document representations; algorithms include agglomerative clustering [Croft, 1980; Griffiths et al., 1986; van Rijsbergen and Sparck Jones, 1973], partition-based clustering [Hearst and Pedersen, 1996], latent semantic analysis [Deerwester et al., 1990], and several language-model based techniques [Xu and Croft, 1999; Hofmann, 1999; Liu and Croft, 2004; Kurland, 2006; Wei and Croft, 2006].

In hard clustering, the documents in the collection are partitioned in k topic-based clusters. We can represent a clustering as the $k \times n$ matrix, $\mathbf{V}$, where columns are binary vectors indicating the cluster membership of each document. The classic example of a hard clustering is the hard k-means algorithm.

Hard clustering techniques limit a document to a single, discrete cluster label. However, documents rarely discuss one topic. Because of this, several clustering methods exist based on assigning documents to multiple clusters. *Soft clustering* refers to assigning each document



(a) Good Separation



(b) Bad Separation

Figure 3.2. Artificial scenarios where the cluster hypothesis holds (a) and does not hold (b).

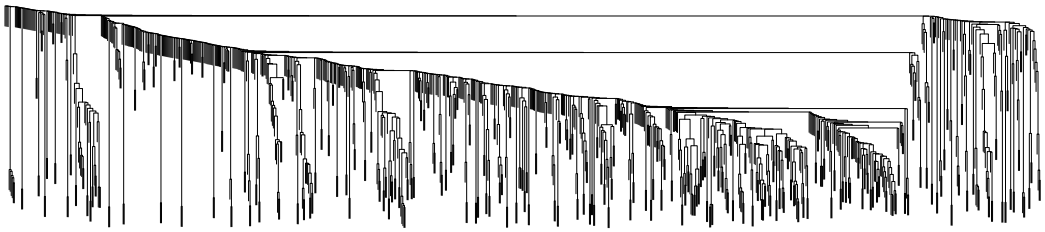
Agglomerative Clustering

1. Compute $n \times n$ affinity matrix,**A**
2. Find the entry with the highest similarity, A_{ij}
3. Compute the similarity between all clusters and the cluster formed by merging i and j Place similarities in $A_{i,\cdot}$, $A_{\cdot,i}$
4. Remove $A_{j,\cdot}$, $A_{\cdot,j}$
5. If **A** is of more than 1 dimension, goto 2

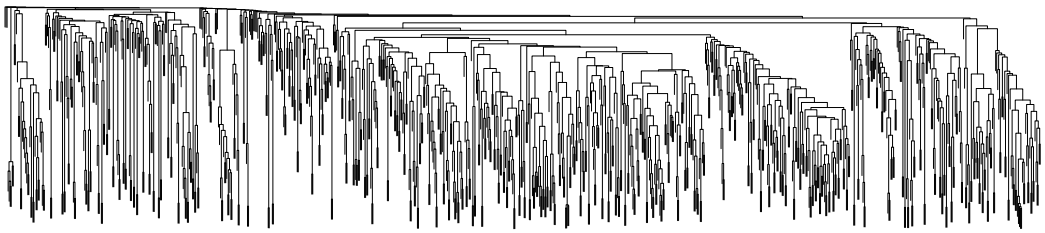
Figure 3.3. General Agglomerative Clustering Algorithm.

to a set of clusters. In the vector space model, a clustering can be produced by projecting documents into a lower-dimensional space derived by, for example, soft k-means or singular value decomposition. The matrix $\mathbf{V}$ now contains real valued components such that $V_{i,j}$ refers to the degree to which the document j discusses a topic i. In language modeling, clusters are frequently referred to as topics and documents are represented as mixtures of topic language models. So, probabilistic semantics are attached to $\mathbf{V}$ so that $V_{i,j}$ refers to the probability that a document discusses a topic, $P(z = i|d = j)$ where z is a random variable over k topics and d is a random variable over documents.

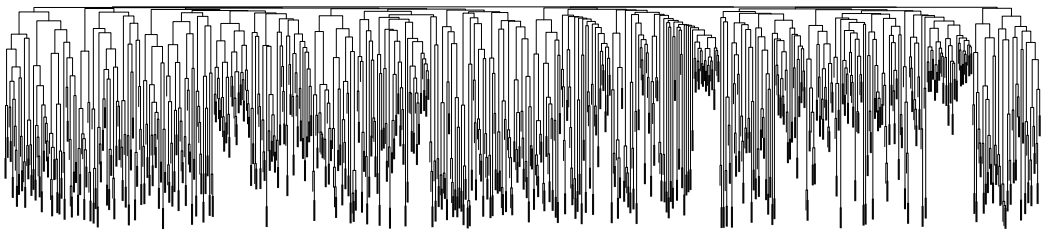
One drawback of both hard and soft clustering is that partitions are considered independent of each other. However, inter-topic relationships are often assumed to be hierarchical; i.e., topics are composed of subtopics, subtopics are composed of subsubtopics, *et cetera*. This hierarchical structure can be modeled by using *agglomerative clustering*. Agglomerative clustering iteratively builds a hierarchy of clusters by merging similar documents and clusters. We present the general agglomerative clustering algorithm in Figure 3.3. When constructing a hierarchy, we are free to select a suitable agglomeration method (Step 3). Three popular agglomeration methods include single link, average link, and Ward's method. Single link agglomeration refers to computing the similarity between two clusters i and j as the highest similarity between pairs of documents spanning i and j. Average link agglomeration refers to computing the similarity between two clusters i and j as the average similarity between pairs of documents spanning i and j. Ward's method refers to computing the similarity between two clusters i and j as the variance-weighted average similarity between pairs of documents spanning i and j. Agglomerative clustering results in a hierarchy referred to as a *dendrogram*. We present example dendrograms in Figure 3.4. Dendrograms are full binary trees. Therefore, there are $k = n - 1$ non-singleton clusters and the cluster assignment matrix, $\mathbf{V}$, has dimension $(n - 1) \times n$. In general, single link clustering tends to result in long "straggly" clusters, while average link and Ward's method tend to produce more compact. Ward's method, because of the variance weighting, creates elliptical clusters with a relatively flatter hierarchy.



(a) single link



(b) average link



(c) Ward

Figure 3.4. Dendrograms for three hierarchical clustering methods. The top 1000 documents retrieved for the query "hostage-taking" were hierarchically clustered according to single link, average link, and Ward's method.

3.1.1.2 Cluster-based Retrieval

If we assume that a single, distinct cluster of relevant documents exists for a query, then we may want to develop algorithms that find and retrieve documents from the relevant cluster and no others. This is one type of *cluster-based retrieval*.

An ancillary data structure which is often produced from a clustering algorithm is an $m \times k$ matrix, $\mathbf{U}$, which represents the clusters in the ambient space. In the vector space model, these are often produced by averaging the ambient representations of the member documents. These averages are referred to as the cluster *centroids*. In the language modeling framework, the columns of $\mathbf{U}$ are multinomials referred to as topic language models (i.e., $U_{i,j} = P(w = i|z = j)$). Other approaches include representing clusters by a single document typical of the cluster. Typicality here could refer to the document in the densest region of the cluster. These documents are referred to as the cluster *mediods*. The matrix $\mathbf{U}$ allows us to score clusters. In the vector space model, we can rank each cluster i according to $\cos(\mathbf{U}_{\cdot,i}, \mathbf{q})$. Using language models, we rank each cluster i according to $\langle \log(\mathbf{U}_{\cdot,i}), \mathbf{q} \rangle$.

When ranking clusters, Jardine and van Rijsbergen found that incorporating the size of the cluster into ranking function improved effectiveness [1971]. Specifically, Jardine and van Rijsbergen use binary term vectors where the component value is 1 if the within-cluster document frequency is greater than $\log |C|$ where $|C|$ is the size of the cluster. Croft used scalar indexing by computing smoothed, within-cluster document relative frequencies [1980].

We can use cluster scoring in order to guide the retrieval process. The simplest retrieval method considers only the top-ranked cluster. Let c represent the top-ranked cluster. We can then retrieve documents according to,

1. the *set* defined by $\{i|V_{c,i} > 0\}$
2. the *ranking* of documents in $\{i|V_{c,i} > 0\}$ according to $\mathbf{d}^{\mathsf{T}}\mathbf{q}$
3. the *ranking* defined by the row $V_{c,\cdot}$

Early work demonstrated that using this technique with hard clustering hurt effectiveness; retrieved documents often included many non-relevant documents [Salton, 1971]. In the context of single link hierarchical clustering, Jardine and van Rijsbergen showed that ranking all k clusters and retrieving a set of documents improved the effectiveness of search over non-cluster techniques for high precision evaluation [1971].

If we treat each non-leaf node of the dendrogram as a retrievable cluster, then we can exploit the hierarchy in order to search for the top-ranked cluster. Jardine and van Rijsbergen proposed searching for the top-ranked cluster by scoring clusters starting at the root of the dendrogram and stopping when scores stopped increasing [1971]. We show this graphically in Figure 3.5. This demonstrated effectiveness similar to performing a global search for the top-ranked cluster.

These results in top-down search used the size-penalized version of cluster-ranking biasing retrieved clusters toward those which occur near the bottom of the hierarchy. Croft proposed a bottom-up cluster search method which ranks the set of clusters with any leaf children [1980]. This method outperformed top-down searches and outperformed non-cluster techniques for high precision evaluation.

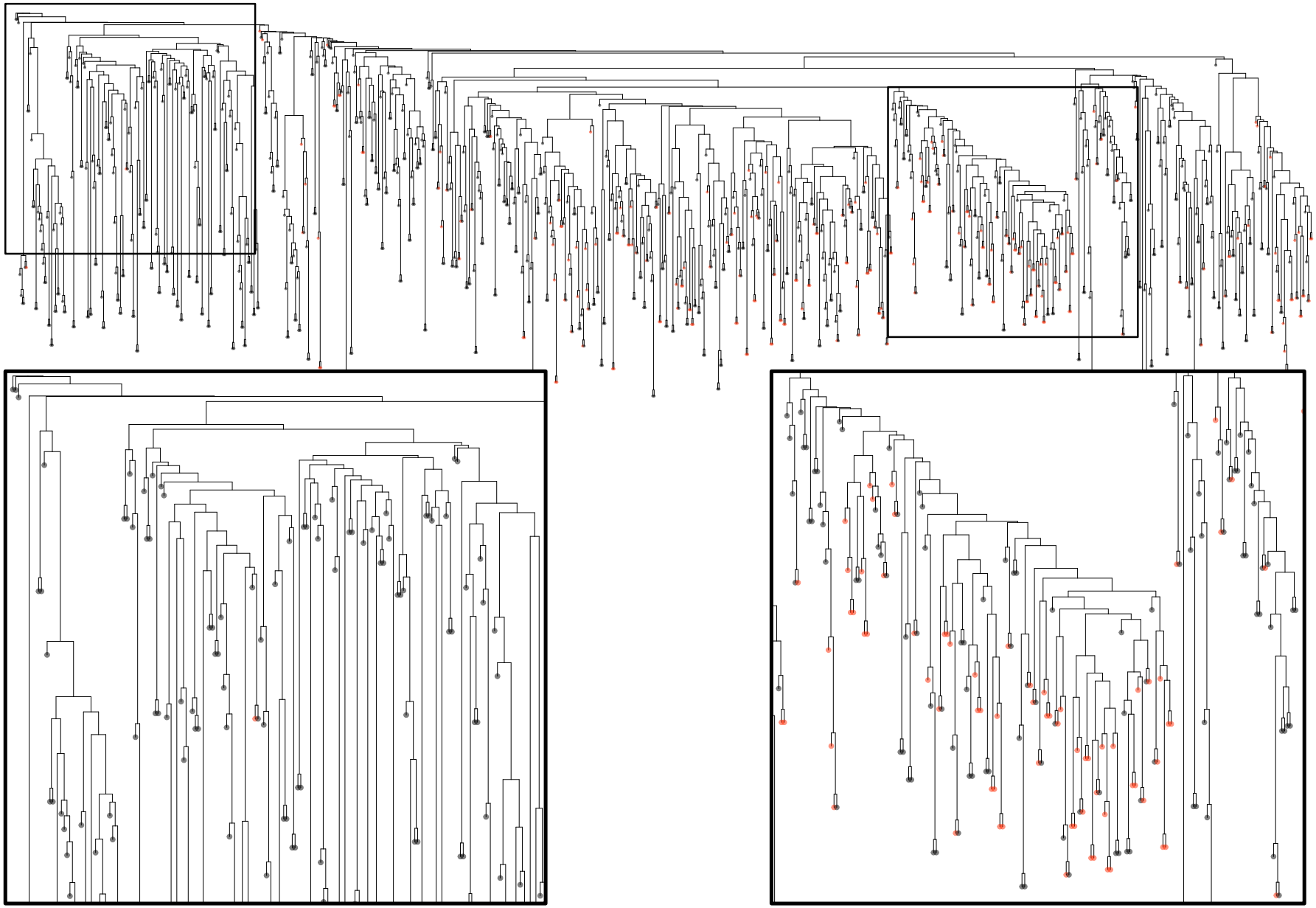


Figure 3.5. Hierarchical cluster-based retrieval. In top-down retrieval, the top-scoring cluster is found by moving down from the root of the dendrogram until cluster scores stop increasing. In bottom-up retrieval, the the top-scoring cluster is found by moving up from the bottom non-leaf nodes of the dendrogram.

The success of bottom-up searches may be alarming because it suggests two possible issues with hierarchical clustering. First, matching queries against higher-level clusters may not adequately incorporate information important to the performance measure. For example, consider the representation of a cluster containing a pair of closely-related relevant documents. In the vector space model, this cluster might be represented at retrieval time—as opposed to clustering time—by the linear combination of component document vectors. The introduction of random, non-relevant documents will not, on average, affect the centroid. But these are precisely the types of clusters we expect to find at the top of the hierarchy. Since both centroids are the same, and since higher level clusters may contain non-relevant documents, we may expect queries matching lower level clusters to perform better. A second observation suggested by the success of bottom-up searches is that the agglomeration process is unsatisfactory. Regardless of how well we represent clusters of documents, if the underlying process does not group together similar documents, then effective cluster-based retrieval is very difficult.

These concerns were partially addressed by the work of Griffiths *et al.* which demonstrated that different agglomeration methods could be used to address the problems with single link clusters [1986]. In the context of bottom-up searches, agglomeration by average link and Ward's method were shown to consistently outperform single link clusters. We can refer to Figure 3.4 to better understand the difference between single link clusters and these other two methods. Clusters in the the single link hierarchy are taller, illustrating the greedier fashion in which clusters are formed. Recall also that this greedy clustering results in straggly clusters that potentially span relatively large areas of space. Average link and Ward's method clusters are much shorter, indicating that they tend to be localized in compact regions of space. The success of these localized methods may indicate that relevant documents tend to be isolated to small areas of space, as opposed to large or stringy areas.

3.1.1.3 Document Expansion

Document vectors, despite being estimated from long text samples, still suffer from sparsity. When the sparsity is systematic within the topic of the document we run the risk of missing the document during retrieval. For example, if the author never uses the word "canine" in a document about "dogs", then the system will always miss this document when a user queries "dog". Salton recognized that a document's representation should include terms occurring in topically-related documents [Salton, 1968],

> The value of a document is assumed to be a linear function of the values of the terms it contains as well of the values of the associated documents.

We refer to this process as *document expansion*. When the topical relationships are based on clusters, we refer to this process as *cluster-based document expansion*. Originally proposed in the context of language modeling [Liu and Croft, 2004], an expanded document representation can be formulated as,

$$P(w|\theta_d) = \lambda P(w|\theta_d) + (1-\lambda)\sum_{i=1}^{k} P(w|\theta_i)c(d,i) \tag{3.1}$$

where $c(d, i)$ is an indicator function whose value is 1 if d belongs to cluster i and θ_i is the topic model i. When documents are composed of multiple topics, Wei and Croft [Wei and Croft, 2006] proposed the following expanded document model,

$$P(w|\theta_d) = \lambda P(w|\theta_d) + (1 - \lambda) \sum_{i=1}^{k} P(w|\theta_{c_i}) P(z = i|d) \tag{3.2}$$

where $P(z = i|d)$ is derived from $\mathbf{U}$, computed by using Blei and Lafferty's Latent Dirichlet Allocation clustering technique [Blei et al., 2003]. Hoffman [Hofmann, 1999] studied this expansion method the special case of $\lambda = 0$ when using the Probabilistic Latent Semantic Analysis clustering approach.

3.2 The Voorhees Test

Voorhees, concerned with the disparity between the size of RR and NR in Figure 3.2, suggested an alternative test [Voorhees, 1985]. Instead of looking at the distribution of similarities, Voorhees measured the density of relevant documents near other relevant documents. That is, for each relevant document, we will look at its k nearest neighbors and compute what we will refer to as the *local precision*. For example, if $k = 5$, for each relevant document, we look at its five closest documents; the local precision is the number of relevant documents divided by five. Voorhees then tests the following hypothesis

Hypothesis 3.2. *Relevant documents have high local precision.*

Graphic representations of local precision distributions originally presented in [Voorhees, 1985] are displayed in Figure 3.6. By qualitative inspection, Voorhees argued that the MED collection satisfies Hypothesis 3.2 because relevant documents tend to be related to other relevant documents. Relevant documents in the CACM, CISI, and INSPEC collections in Figure 3.6, however, tend to be isolated from each other, implying that Hypothesis 3.2 is not supported for these three collections.

3.2.1 Approaches Motived by the Voorhees Test

The Voorhees test has received much less attention in the information retrieval literature than the Jardine-van Rijsbergen test. In this section, we will review several approaches which assume that queries satisfy Hypothesis 3.2.

3.2.1.1 Multiple-Cluster Retrieval

The Jardine-van Rijsbergen test motivated retrieving a single cluster from some set of clusters. If searched top-down, then retrieved clusters may include many non-relevant documents. If searched bottom-up, then retrieved clusters may have higher precision but lower recall. The Voorhees test suggests that relevant documents potentially occur in isolated, locally-dense clusters. Recall that we can capture the local density by searching bottom-up, as suggested by Croft. We can retrieve disparate clusters by simply retrieving and merging documents from multiple clusters instead of just one. Voorhees proposed ranking the bottom level single link clusters in a dendrogram and retrieving the top-ranking document from

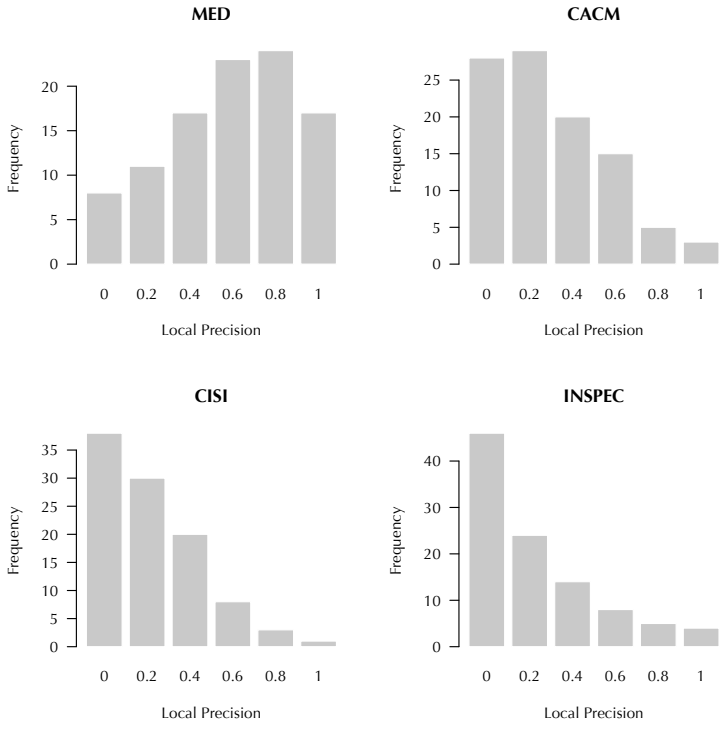


Figure 3.6. Local precision cluster hypothesis test for four collections presented in [Voorhees, 1985]. For each relevant document, we compute the number of relevant documents in its five nearest neighbors; we refer to this as the local precision. According to this measure, the MED collection exhibits high clustering; relevant documents tend to be near other relevant documents. On the other hand, relevant documents in other collections tend be surrounded by fewer relevant documents

each cluster [1985]. This approach improved the effectiveness of the retrieved set of documents. Griffiths *et al.* demonstrated similar improvements when retrieving multiple average link and Ward's method clusters [1986].

One conclusion we can draw from the increasing effectiveness as we move from top-down to bottom-up, from single link to Ward's method, and from single cluster to multiple cluster retrieval is that relevance tends to be supported by many, small, tight clusters as opposed to larger, stragglier clusters. Griffiths *et al.* took this to an extreme and considered clusters consisting of only pairs of nearest-neighbors [1986]. This algorithm performed multiple cluster retrieval using clusters consisting of pairs of documents. Not only did this clustering outperform all other cluster-based retrieval methods but it also outperformed non-cluster techniques for some collections. These results were confirmed in the context of language modeling [Kurland and Lee, 2004].

3.2.1.2 Spreading Activation

Griffiths *et al.* took cluster-based retrieval to one extreme by retrieving small nearest-neighbor clusters. The success of this representation and algorithm demonstrates a progression toward increasingly local analysis of relevance. In this section, we will describe spreading activation, a method which, although predating some of the cluster-based retrieval approaches, can be seen as a philosophical descendent of cluster-based retrieval methods. We believe that spreading activation assumes—though never explicitly tests—the Voorhees test to be true.

Spreading activation refers to the technique of propagating relevance information between topically related documents, represented by the matrix $\mathbf{W}$ defined earlier.[1] Spreading activation usually does no explicit clustering and uses only pairwise relationships between documents. The algorithm is usually initialized by attaching a relevance value to the nodes of the graph either as explicit binary labels or as scores from some initial retrieval. Each node in the graph then recomputes its score by inspecting the relevance values of its neighbors. This process is iterated until convergence. Several propagation rules have been studied in the spreading activation literature. The updated score for a document might be the maximum score of related documents, the average, or some other aggregating function. The relationship to the Voorhees test should be clear. Spreading activation makes the implicit assumption that the relevance label of each node is related to the relevance label of topically related documents.

The original spreading activation model proposed by Preece used manually-built graphs with multiple link types [1981]. The assumption behind manually-built graphs is that the document scores should be correlated by the manual information; certainly we could think of manual labels which would not indicate a correlation. When suitable manual links are absent, topical relationships can be approximated by the methods presented in Section 2.3. Croft's work in network-based collection representations studied combinations of manual and automatic relationships. For example, the I^3R system combined citation and automatic

[1] In general, the graph may consist of heterogeneous nodes including documents, terms, metadata, and concepts. We will focus on homogeneous graphs where nodes correspond to documents and edges correspond to some relationship between documents. For general spreading activation, readers should consult Crestani's survey of classic spreading activation models [Crestani, 1997].

links [Croft et al., 1988]. The results indicate that in many situations, tf.idf baselines could be improved by propagating relevance information over both automatic and citation-based links. The latter were shown to be superior to automatic links. Croft and Turtle extended the description of the I^3R system to consider hypertext as well citations [1989]. Both types of links were demonstrated to improve performance, although hyperlinks provided a larger improvement. Similar experiments using variations of propagation methods, baseline algorithms, and similarity functions have shown similar improvements. More recent results in the context of web search replicate these results for newer collections [Qin et al., 2005; Shakery and Zhai, 2006].

3.2.1.3 Local Document Expansion

Salton originally described document expansion in the context of the vector space model and document vectors were interpolated with nearest-neighbors' vectors. We refer to this process as *local document expansion* because there is no explicit clustering performed. Also operating within the vector space model, Singhal and Pereira applied local document expansion in order to reduce noise in collections of speech-recognized documents [Singhal and Pereira, 1999]. In the language modeling framework, Ogilvie originally proposed nearest-neighbor smoothing [Ogilvie, 2003] while Kurland and Lee rigorously evaluated it [Kurland and Lee, 2004]. This approach has also been used in hypertext collections for propagating term weights across hyperlinks [Qin et al., 2005].

3.3 Jardine-van Rijsbergen or Voorhees?

The critical difference between Hypothesis 3.1 and Hypothesis 3.2 lies in the assumption each makes about the set of relevant documents. Specifically, Hypothesis 3.1 assumes that there is a single, coherent relevant cluster while Hypothesis 3.2 only assumes that relevant documents have high local precision. The implication of this latter assumption can be demonstrated by inspecting the behavior of the measurements as the size of the relevant document set grows for different numbers of relevant clusters. In Figure 3.7, we show artificial 2-dimensional data produced to represent a single relevant cluster in the midst of non-relevant data. Notice that as the size of the relevant cluster grows, both Hypothesis 3.1 and Hypothesis 3.2 receive increasing support. However, if we have several distinct-but-separated clusters, we observe very different behavior. In Figure 3.8, we present artificial data exhibiting four relevant clusters. Because relevant documents do not exist in a single cluster, the RR and NR distributions are difficult to distinguish. This effect is produced because the RR distribution includes distances between documents in different relevant clusters. As the number of relevant clusters (and sample from each) grows, the RR distribution will begin to include more pairs documents with low similarity. Hypothesis 3.2, because it focuses more on the local behavior, is well-supported, regardless of the number of relevant clusters.[2]

[2]The presence of several, distinct clusters is not exceptional. The TREC interactive track, for example, studied queries consisting of several aspects [Over, 1996]. Leouski and Allan, studying interactive retrieval and visualization, noted that the relevant document set is likely to include multiple clusters [Leouski and Allan, 1998].

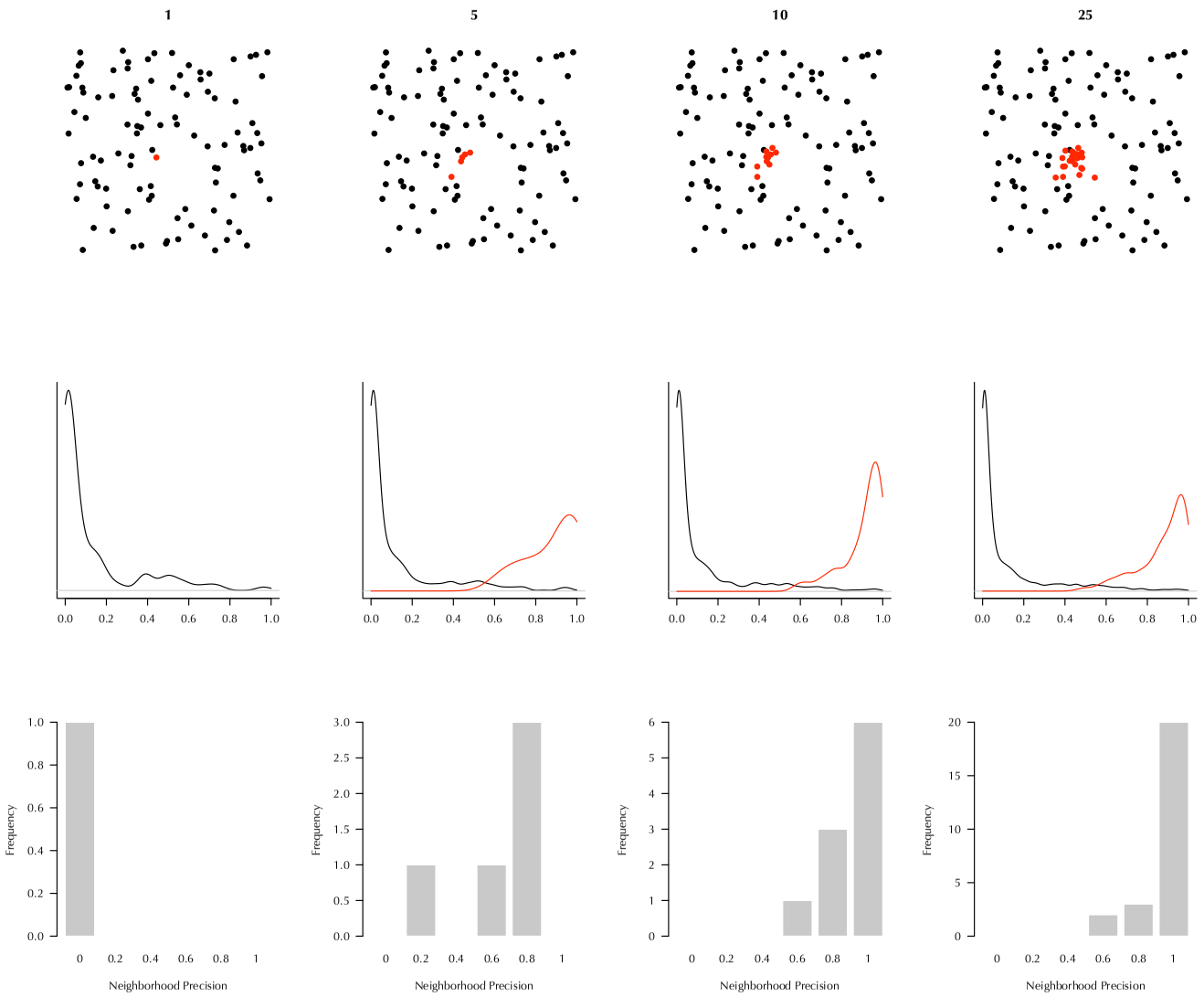


Figure 3.7. Artificial 2-dimensional data produced to represent a single relevant cluster (red points) in the midst of many non-relevant data (black points). The top row shows the relevant cluster developing as the number of relevant points grows from 1 to 25. The second row shows the distributions of similarities between relevant documents (RR) and relevant and non-relevant documents (NR). The third row shows the distribution of local precision. Relevant points are sampled from a Gaussian; non-relevant points are sampled uniformly.

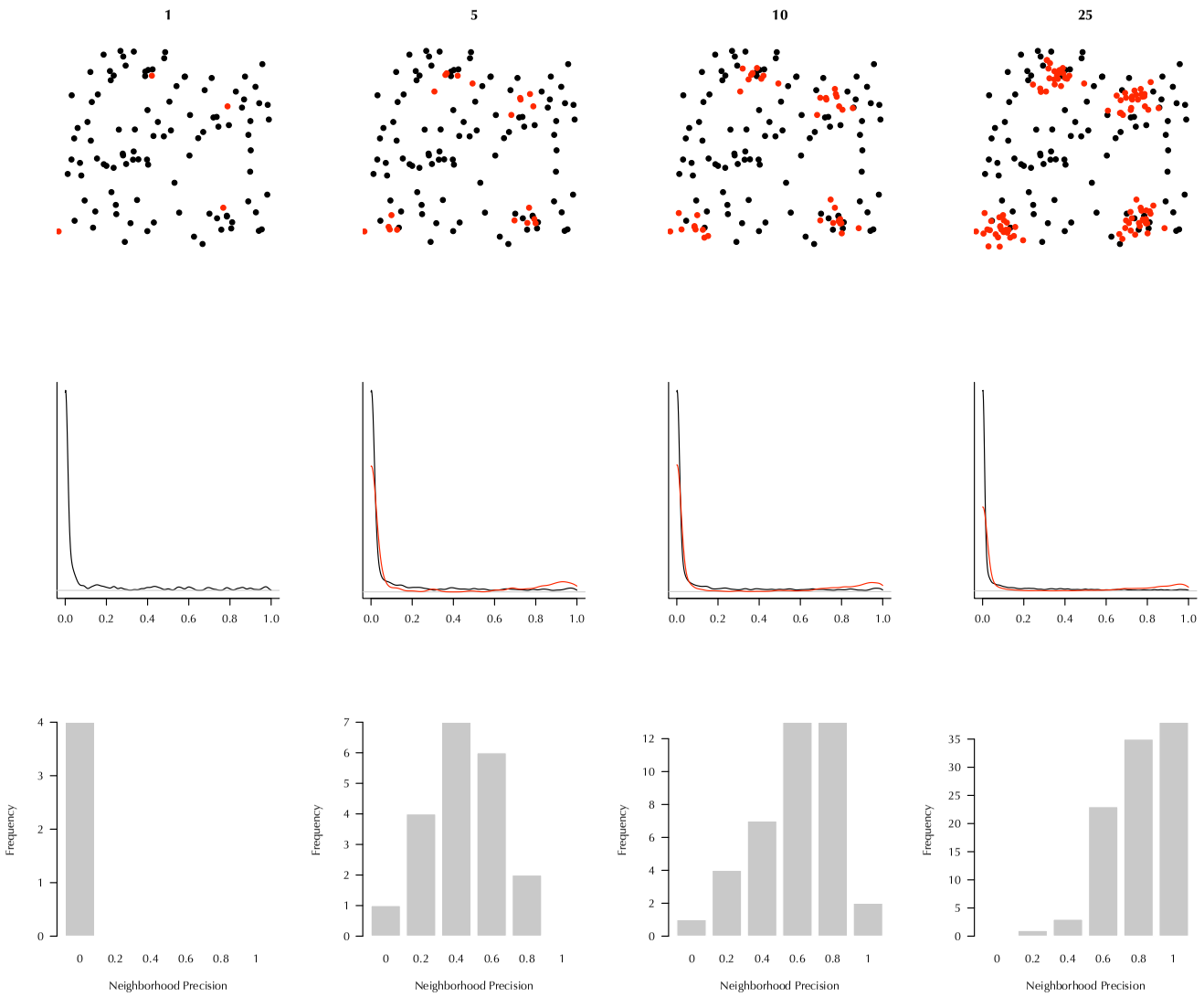


Figure 3.8. Four relevant clusters of varying sizes. The top row shows the relevant cluster developing as the number of relevant points per relevant cluster grows from 1 to 25. The second row shows the distributions of similarities between relevant documents (RR) and relevant and non-relevant documents (NR). The third row shows the distribution of local precision. Relevant points are sampled from a Gaussian; non-relevant points are sampled uniformly.

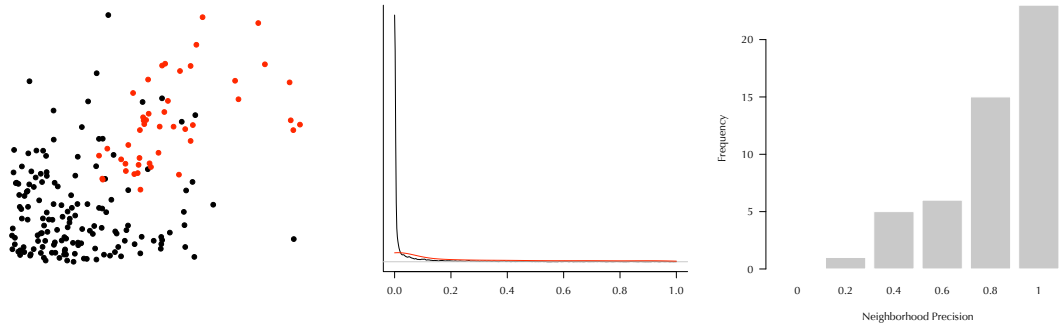


Figure 3.9. Two relevant cluster of non-uniform density. The first subfigure shows fifty relevant points and 150 non-relevant points in two dimensions. The second subfigure shows the distributions of similarities between relevant documents (RR) and relevant and non-relevant documents (NR). The third subfigure shows the distribution of local precision.

Another problem with testing Hypothesis 3.1 occurs when the similarity measure exhibits non-uniform behavior between topically related documents. Consider the artificial example in Figure 3.9. Non-relevant documents exhibit very tight clustering while the relevant set of documents is more diffuse. The effect is that the similarities in the submatrix RR tend to be much larger than the similarities in the submatrix NR. Again, the local precision is more robust to this non-uniformity because the nearest-neighbor criterion is adaptive.

So far, we have used artificial examples when comparing the two methods of testing the cluster hypothesis. Therefore, we performed the same measurements for two sets of queries used in this thesis: ql/trec12 and ql/robust (see Appendix A for details). In these experiments, we microaveraged similarities and local precisions over a set of 150 queries for trec12 and 250 queries for robust. The set of queries represented by trec12 are considered by the community to be relatively easier to satisfy than those in robust. We present our results in Figure 3.10. Qualitatively, we do not see compelling evidence for Hypothesis 3.1 in either collection. Although we would like there to be more evidence for the cluster hypothesis in the easier collection, there also does not seem to be a difference in the overlap of the distributions for the two collections. Hypothesis 3.2, on the other hand, is well-supported for the trec12 collection. The robust collection tends to include many more isolated relevant documents which is consistent with our impression that it is more difficult. Hence, the Voorhees test not only correctly detects clustering in both corpora (more than half of the relevant documents have a local precision of > 0.50), but also distinguishes between these two collections.

3.4 Summary

Vector space representations of text are fundamentally mysterious because of their high-dimensionality. We cannot visually inspect patterns of points in the ambient space. We can try to visualize this by projecting vectors into two or three dimensions but must acknowledge

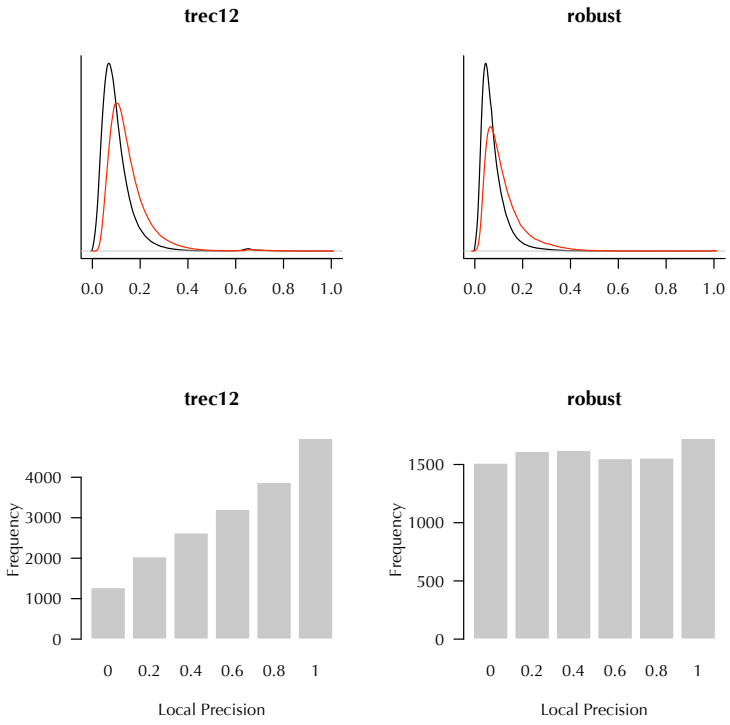


Figure 3.10. Microaveraged values of the Jardine-van Rijsbergen and Voorhees tests for baseline retrievals using the trec12 and robust collections. The Jardine-van Rijsbergen test implies that relevant documents in both collections are poorly-separated from non-relevant documents; it also does not distinguish between the degree of separation between these two collection. The Voorhees measure indicates that the relevant documents in the trec12 collection tend to be related to other relevant documents; this property is not as apparent in the robust collection.

that a huge amount of information is lost in the process. If we want to make statements about the collection, we can only do so by measuring properties of the affinity matrix. Although both the Jardine-van Rijsbergen and Voorhees tests accomplish this, we adopt the Voorhees test because it introduces fewer assumptions about the behavior of relevant documents and the uniformity of the similarity measure. In the next chapter, we will adapt the Voorhees test to measure the local consistency of scores.

CHAPTER 4

AUTOCORRELATION OF RETRIEVAL SCORES

In order to provide evidence for the Voorhees hypothesis, we demonstrated that relevant documents tended to be situated near other relevant documents. In this chapter, we will be relaxing the Voorhees hypothesis to make a statement about retrieval scores. We will be testing the following hypothesis,

Hypothesis 4.1. *Given a set of retrieval scores, the local consistency of the scores correlates positively with retrieval performance.*

In other words, a good retrieval tends to score topically-related documents consistently. We will develop a test which measures the degree to which a retrieval system exhibits local consistency. Like the tests in Chapter 3, our approach will use the inter-document similarity matrix. Unlike these tests, we will use a vector of retrieval scores, $\mathbf{y}$, defined in Section 2.2 instead of relevance judgments.

In this chapter, we will argue that local score consistency is an important predictor of the performance of a set of retrieval scores. Our approach is similar to Fang *et al*'s study of heuristics used in information retrieval [Fang et al., 2004, p. 49; emphasis added],

> *We formally define a set of basic desirable constraints that any reasonable retrieval formula should satisfy, and check these constraints on a variety of retrieval formulas*, which respectively represent the vector space model (pivoted normalization), the classic probabilistic retrieval model (Okapi), and the recently proposed language modeling approach (Dirichlet prior smoothing). We find that none of these retrieval formulas satisfies all the constraints unconditionally, though some formulas violate more constraints or violate some constraints more "seriously" than others. *Empirical results show that when a constraint is not satisfied, it often indicates non-optimality of the method, and when a constraint is satisfied only for a certain range of parameter values, its performance tends to be poor when the parameter is out of the range. In general, we find that the empirical performance of a retrieval formula is tightly related to how well it satisfies these constraints.* Thus the proposed constraints provide a good explanation of many empirical observations about retrieval methods. Moreover, these constraints make it possible to evaluate any existing or new retrieval formula analytically and suggest how we may further improve a retrieval formula.

We will demonstrate that Hypothesis 4.1, like the heuristics studied by Fang *et al*, suggests a property that retrieval systems should incorporate by design.

4.1 Testing the Cluster Hypothesis without Relevance Judgments

There is a growing body of work that studies the correlation of the performance of individual retrievals with the degree to which a retrieval is clustered. In this section, we will review two of these approaches.

4.1.1 Clarity

Clarity measures the extent to which vocabulary is shared in the top $\tilde{n}$ retrieved documents [Cronen-Townsend et al., 2006]. The conjecture is that, in a good retrieval, the most frequent words are topically coherent. A bad retrieval would include documents on many disparate topics; the most frequent terms would be terminological noise.

Clarity measures the similarity of the most frequent words in retrieved documents to the most frequent words used in the whole corpus. We refer to the frequency of terms in the whole corpus as the background frequency. The frequent terms in a good retrieval will be distinct from the background; the frequent terms in a bad retrieval will be similar to the background. In the context of language modeling, we can compute a representation of the language used in the initial retrieval as a weighted combination of document language models,

$$\tilde{\mathbf{q}} = \frac{1}{||\mathbf{y}||_1}\mathbf{C}^\mathsf{T}\mathbf{y} \tag{4.1}$$

where $\mathbf{y}$ represent the document query likelihoods (Equation 2.11). In order to model "general text", we use corpus-level statistics. The assumption here is that a language model of the entire corpus will naturally converge on non-specific terminology.

$$\mathbf{c} = \frac{1}{\|\mathbf{C}^\mathsf{T}\mathbf{e}\|_1}\mathbf{C}^\mathsf{T}\mathbf{e} \tag{4.2}$$

We can compare $\tilde{\mathbf{q}}$ with $\mathbf{c}$ using any of the methods from Section 2.3.2. For example, we can use the Kullback-Leibler divergence, $D_{KL}(\tilde{\mathbf{q}}\|\mathbf{c})$, or the Jensen-Shanon divergence, $JS_\pi(\tilde{\mathbf{q}}, \mathbf{c})$.

When retrievals are not based on language modeling, Equation 4.1 can be adjusted to be a function of document ranks instead of scores [Cronen-Townsend et al., 2006]. Ranked-list Clarity converts document ranks to $P(Q|\theta_i)$ values. This conversion begins by replacing all of the scores in $\mathbf{y}$ with the respective ranks. Our estimation of $P(Q|\theta_i)$ from the ranks is,

$$\hat{\mathbf{y}} = \begin{cases} \frac{2(c+1-y_i)}{c(c+1)} & \text{if } y_i \leq c \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where c is a cutoff parameter. We estimate the query language model using Equation 4.1.

4.1.2 Cox-Lewis Statistic

Assume we have a set of documents retrieved for our query and $\tilde{\mathcal{R}}$ represents the indexes of the top $\tilde{n}$ documents. Another way to quantify the dispersion of a set of documents is to

inspect the similarity between documents in $\tilde{\mathcal{R}}$. In spatial data analysis, the Cox-Lewis statistic measures the expected distance between a group of points. In the case of the set $\tilde{\mathcal{R}}$, distances are computed in the m-dimensional embedding space. We hypothesize that a good retrieval will return a single, tight cluster. A poorly performing retrieval will return a loosely related set of documents covering many topics. The method of quantifying this dispersion is to measure the distance from a random document a to its nearest neighbor, b. A retrieval which is tightly clustered will, on average, have a low distance between a and b; a retrieval which is less tightly-closed will, on average have high distances between a and b. This average corresponds to using the Cox-Lewis statistic to measure the randomness of the top $\tilde{n}$ documents retrieved from a system [Vinay et al., 2006]. It is important to notice that the Cox-Lewis statistic throws away information about the retrieval function $\mathbf{y}$. This makes the Cox-Lewis statistic highly-dependent on selecting the top $\tilde{n}$ documents.

4.2 Autocorrelation

In this section, we will be deriving a measure of local score consistency from the Voorhees test. The Voorhees test computes the local precision of each relevant document. Let $\mathbf{r} \in \{0,1\}^n$ be the vector of relevance judgments and $\mathbf{W}$ be the nearest-neighbor matrix defined in Section 2.3.3 where rows are L_1 normalized such that $\mathbf{We} = \mathbf{e}$. We can construct the Voorhees histogram by computing the vector $\mathbf{Wr}$ and then inspecting the entries corresponding to relevant documents. The histogram used in the Voorhees test provides a nice visualization of the distribution of local precision but can be summarized by a single number. For example, we can compute the mean of the local precisions of relevant documents. Noticing that $\mathbf{r}^\mathsf{T}\mathbf{r}$ represents the number of relevant documents, the mean can be computed as

$$\frac{\mathbf{r}^\mathsf{T}\mathbf{W}\mathbf{r}}{\mathbf{r}^\mathsf{T}\mathbf{r}} = \frac{\sum_{i,j} W_{ij} r_i r_j}{\sum_i r_i^2} \tag{4.4}$$

and is more generally referred to as the Rayleigh quotient in mathematics. We are interested in measuring the similarity between the scores in the absence of relevance information. Our approach will be to replace the binary vector, $\mathbf{r}$, with the score vector $\mathbf{y}$. Under the same row normalization assumption of Equation 4.4,

$$\frac{\mathbf{y}^\mathsf{T}\mathbf{W}\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}} = \frac{\sum_{i,j} W_{ij} y_i y_j}{\sum_i y_i^2} \tag{4.5}$$

which is referred to as the Moran autocorrelation in spatial data analysis [Cliff and Ord, 1973; Griffith, 2003].

For arbitrary $\mathbf{y}$ and fixed $\mathbf{W}$, the Rayleigh quotient is bound by above by the largest eigenvalue of $\mathbf{W}$. However, recall that we will be computing a different $\mathbf{W}$ for each retrieval using the top $\tilde{n}$ documents. Therefore, the expected range of the value in Equation 4.5 is dependent on the matrix $\mathbf{W}$ and vector $\mathbf{y}$. This is problematic since we would like to com-

pare autocorrelation values for different retrievals. Therefore, we use the Cauchy-Schwartz inequality to establish a bound on the autocorrelation,

$$\frac{\mathbf{y}^\mathsf{T}\mathbf{W}\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}} \leq \sqrt{\frac{\mathbf{y}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{y}}{\mathbf{y}^\mathsf{T}\mathbf{y}}}$$

Dividing Equation 4.5 by this bound, we define the normalized spatial autocorrelation as

$$I_M = \frac{\mathbf{y}^\mathsf{T}\mathbf{W}\mathbf{y}}{\sqrt{\mathbf{y}^\mathsf{T}\mathbf{y} \times \mathbf{y}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{W}\mathbf{y}}} \tag{4.6}$$

where we adopt the standard notation, I_M, for the Moran autocorrelation.

In Section 2.3.3, we indicated that the nearest-neighbor matrix, $\mathbf{W}$, could be visualized as a graph. A set of scores, $\mathbf{y}$, can be represented by coloring each node of the graph according to its score. We present examples of document graphs with score-based coloring in Figure 4.1. In order to accent the locality of scores, we also colored edges with a gradient which transitions between the colors associated with the scores of the nodes on either end of the edge. The graph of a retrieval with high autocorrelation (Figure 4.1a) consists of edges with more solid colors, resulting in clear contrast between high-scoring and low-scoring regions of the graph. The graph of a retrieval with low autocorrelation (Figure 4.1b) consists of edges with sharper gradients, resulting muddier contrast between high-scoring and low-scoring regions of the graph.

4.3 Experiments

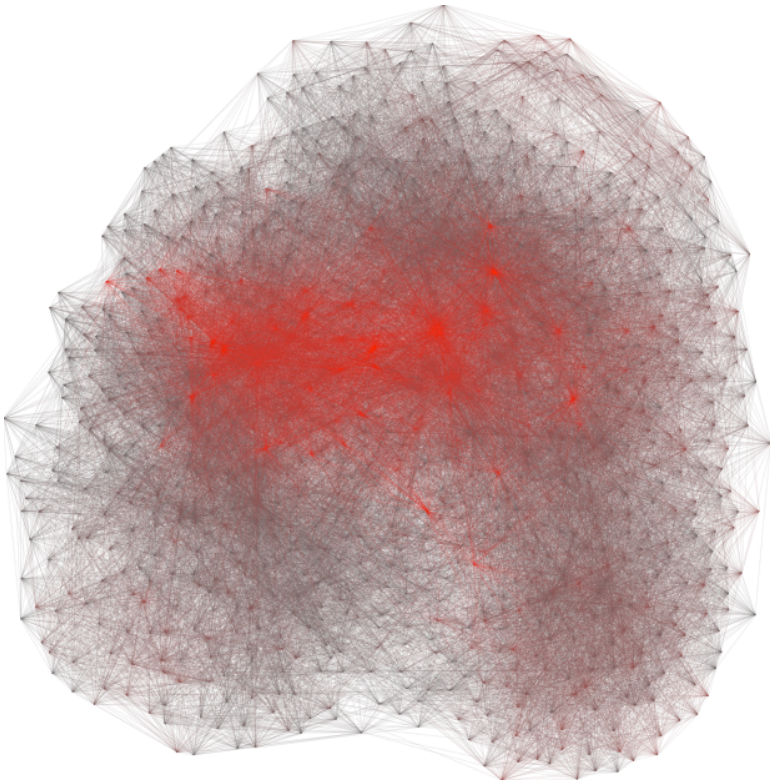
Our experiments focus on testing the ability of autocorrelation to predict the performance of a retrieval. As stated in the introduction of this chapter, we are interested in predicting the performance of the retrieval generated by an arbitrary system. Our methodology is consistent with previous research in that we predict the *relative performance* of a retrieval by comparing a ranking based on our predictor to a ranking based on performance as measured by average precision.

We present results for two sets of experiments. The first set of experiments presents detailed comparisons of our predictors to previously-proposed predictors using their data sets. Our second set of experiments demonstrates the generalizability of our approach to arbitrary retrieval methods, corpus types, and corpus languages.

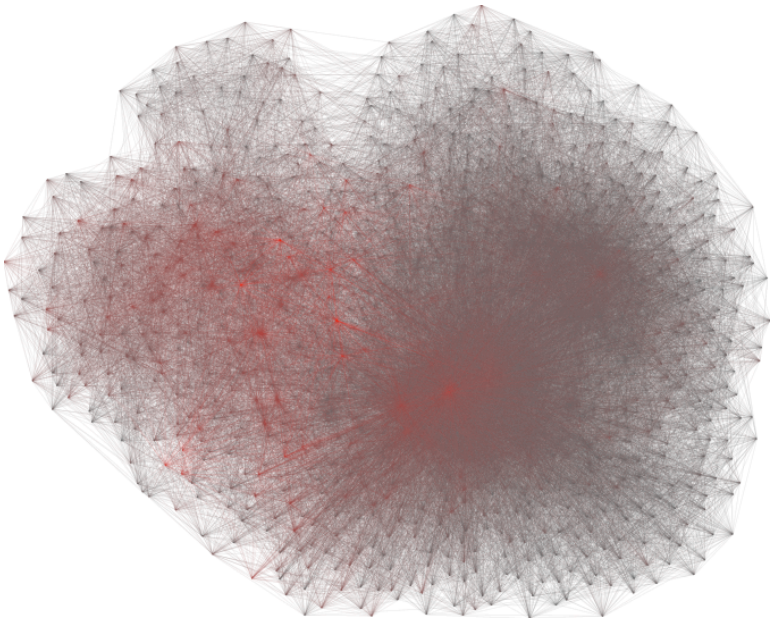
4.3.1 Detailed Experiments

In our detailed experiments, we will predict the performance of language modeling scores using our autocorrelation predictor. We use retrievals, values for baseline predictors, and evaluation measures reported in previous work [Zhou and Croft, 2006]. This will allow us to compare the magnitude of our correlations with previously-published results.

These performance prediction experiments use language model retrievals performed for queries associated with collections in the TREC corpora. Using TREC collections allows us to confidently associate an average precision with a retrieval. In these experiments, we use



(a) a retrieval with high autocorrelation ("dismantling Europe's arsenal")



(b) a retrieval with low autocorrelation ("Export Controls Cryptography")

Figure 4.1. Retrieval functions on the document graph. We constructed a nearest-neighbor document graph for the top 1000 documents from a retrieval. Edges were colored by a gradient based on the relevance of each connected document. High retrieval scores are associated with red. Low retrieval scores are associated with grey.

the following topic collections: TREC 4 ad hoc, TREC 5 ad hoc, Robust 2004, Terabyte 2004, and Terabyte 2005.

We provide two baselines. Our first baseline is the classic Clarity predictor presented in Equation 4.1. Clarity is the theoretically-appropriate predictor for language modeling systems. Our second baseline is Zhou and Croft's "ranking robustness" predictor [Zhou and Croft, 2006]. This predictor corrupts the top k documents from retrieval and re-computes the language model scores for these corrupted documents. The value of the predictor is the Spearman rank correlation between the original ranking and the corrupted ranking. In our tables, we will label results for Clarity using $D_{KL}^{\mathcal{V}}$ and the ranking robustness predictor using P.

In addition to single-predictor experiments, we experimented with the linear combination of predictors. We optimized the linear regression using the square root of each predictor. We found that this substantially improved fits for all predictors, including the baselines. We considered linear combinations of pairs of predictors (labeled by the components) and all predictors (labeled as β).

4.3.2 Generalizability Experiments

Autocorrelation does not require a particular baseline retrieval system; the predictor can be computed for an arbitrary retrieval, regardless of how scores were generated. In a second set of experiments, we demonstrate the generalizability of our results to a variety of collections, topics, and retrieval systems.

We gathered a diverse set of collections from TREC corpora. We cast a wide net in order to locate collections where our predictors might fail. Our hypothesis is that topically-related documents should have similar scores. Therefore, we avoided collections where scores were unlikely to be correlated (eg, question-answering) or were likely to be negatively correlated (eg, diverse ranking). Nevertheless, our collections include corpora where correlations are weakly justified (eg, non-English corpora) or not justified at all (eg, expert search). Details of these corpora and runs can be found in Appendix A.

4.3.3 Evaluation

Given a set of retrievals, potentially from a combination of queries and systems, we measure the correlation of the rank ordering of this set by the predictor and by the performance metric. In order to ensure comparability with previous results, we present the correlation between the predictor's ranking and ranking based on average precision of the retrieval. We present results for Kendall's τ, Spearman's ρ, and Pearson's r. Unless explicitly noted, all correlations are significant with $p < 0.05$.

Predictors can sometimes perform better when linearly combined [Diaz and Jones, 2004; He and Ounis, 2004]. Although previous work has presented the coefficient of determination (R^2) to measure the quality of the regression, this measure cannot be reliably used when comparing slight improvements from combining predictors. Therefore, we adopt the adjusted coefficient of determination which penalizes models with more variables. The adjusted R^2 allows us to evaluate the improvement in prediction achieved by adding a parameter but loses the statistical interpretation of R^2. We, therefore, will use the correlation

coefficients to evaluate the magnitude of the correlation and the adjusted R^2 to evaluate the combination of variables.

4.4 Results

We present results for our detailed experiments comparing the prediction of language model scores in Table 4.1. Although the Clarity measure is theoretically designed for language model scores, it consistently underperforms our system-agnostic predictor. The ranking robustness measure, developed by Zhou and Croft to improve Clarity's performance on web collections (i.e., terabyte04, terabyte05), does improve the τ correlation from 0.139 to 0.150 for terabyte04 and 0.171 to 0.208 for terabyte05. However, these improvements are slight compared to the performance of autocorrelation on these collections. Our predictor achieves a τ correlation of 0.454 for terabyte04 and 0.383 for terabyte05. Though not always the strongest, autocorrelation achieves correlations competitive with baseline predictors. When examining the performance of linear combinations of predictors (Table 4.2), we note that in every case, autocorrelation factors as a necessary component of a strong predictor. We also note that the adjusted R^2 for individual baselines are always improved by incorporating autocorrelation.

We present our generalizability results in Table 4.3. For every collection except one, we achieve better correlations than ranked-list Clarity. Surprisingly, we achieve relatively strong correlations for Spanish and Chinese collections despite our naïve processing. We do not have a ranked-list Clarity correlation for ent05 because we did not have a clear method for building a query-independent language model for an entity. However, our autocorrelation measure does not achieve high correlations perhaps because relevance for entity retrieval does not propagate according to the cooccurrence links we use.

As noted above, the poor Clarity performance on web data is consistent with our findings in the detailed experiments. Clarity also notably underperforms for several news corpora (trec5, trec7, and robust04). On the other hand, autocorrelation seems robust to the changes between different corpora.

4.5 Discussion

We present the results presented in Section 4.4 to provide evidence for Hypothesis 4.1. Our experiments demonstrate that a failure to respect the local consistency correlates with poor performance. Why might systems fail to score topically-related documents consistently? Query-based information retrieval systems often score documents independently. All of the retrieval models in Chapter 2 score documents independently. That is, the score of document a may be computed by examining query term or phrase matches, the document length, and perhaps global collection statistics. Once computed, though, a system rarely compares the score of a to the score of a topically-related document b. Our results demonstrate that, when absent, attention to local score consistency can hurt performance.

(a) Kendall's τ

	$D_{KL}^{\mathcal{V}}$	P	I_M
trec4	0.353	**0.548**	0.513
trec5	0.311	0.329	**0.357**
robust04	**0.418**	0.398	0.373
terabyte04	0.139	0.150	**0.454**
terabyte05	0.171	0.208	**0.383**

(b) Spearman's ρ

	$D_{KL}^{\mathcal{V}}$	P	I_M
trec4	0.507	**0.738**	0.674
trec5	0.447	0.475	**0.498**
robust04	**0.590**	0.567	0.543
terabyte04	0.193	0.221	**0.583**
terabyte05	0.246	0.307	**0.522**

(c) Pearson's r

	$D_{KL}^{\mathcal{V}}$	P	I_M
trec4	0.430	0.613	**0.645**
trec5	0.366	0.454	**0.538**
robust04	0.509	**0.554**	0.349
terabyte04	0.305	0.341	**0.598**
terabyte05	0.206	0.301	**0.539**

Table 4.1. Comparison of autocorrelation to Robustness and Clarity measures for language model scores. Evaluation replicates experiments from [Zhou and Croft, 2006]. We present correlations between the classic Clarity measure ($D_{KL}^{\mathcal{V}}$), the ranking robustness measure (P), and autocorrelation (I_M) each with mean average precision. Measures in bold represent the strongest correlation for that test/collection pair.

	$D_{KL}^{\mathcal{V}}$	P	I_M	$D_{KL}^{\mathcal{V}}, P$	$D_{KL}^{\mathcal{V}}, I_M$	P, I_M	β
trec4	0.168	0.363	0.422	0.466	0.420	**0.557**	0.553
trec5	0.116	0.190	0.236	0.238	0.244	0.266	**0.269**
robust04	0.256	0.304	0.278	0.403	0.373	0.402	**0.442**
terabyte04	0.059	0.045	0.292	0.076	**0.293**	0.289	0.284
terabyte05	0.022	0.072	0.193	0.120	0.225	0.218	**0.257**

Table 4.2. Combination of autocorrelation, ranking robustness, and Clarity measures for language model scores. The adjusted coefficient of determination is presented to measure the effectiveness of individual predictors, pairwise combinations of predictors, and the combination of all predictors (β). Measures in bold represent the strongest correlation for that test/collection pair.

	D_{KL}	I_M
trec3	0.201	**0.461**
trec4	0.252	**0.396**
trec5	0.016	**0.277**
trec6	**0.230**	0.227
trec7	0.083	**0.326**
trec8	0.235	**0.396**
robust03	0.302	**0.354**
robust04	0.183	**0.308**
robust05	0.224	**0.249**
terabyte04	0.043	**0.245**
terabyte05	0.068	**0.306**
trec4-spanish	0.307	**0.388**
trec5-spanish	0.220	**0.458**
trec5-chinese	0.092	**0.199**
trec6-chinese	0.144	**0.276**
ent05	-	0.181

Table 4.3. Predicting the ranking of large sets of retrievals for various collections and retrieval systems. Kendall's τ correlations are computed between the predicted ranking and a ranking based on the retrieval's average precision. Measures in bold represent the strongest correlation for that test/collection pair.

In Equation 4.6, we presented the Moran autocorrelation measuring the local consistency. We can rewrite this equation as the correlation between two vectors,

$$I_M = \frac{\mathbf{y}^\mathsf{T}\tilde{\mathbf{y}}}{\|\mathbf{y}\|_2\|\tilde{\mathbf{y}}\|_2} \tag{4.7}$$

where $\tilde{\mathbf{y}} = \mathbf{W}\mathbf{y}$. This implies that a vector of scores, $\mathbf{y}$, has high autocorrelation if it is correlated with the vector $\tilde{\mathbf{y}}$. This vector, $\tilde{\mathbf{y}}$, can be interpreted as the original set of retrieval scores "diffused" over the adjacency graph, $\mathbf{W}$. From another perspective, the vector $\tilde{\mathbf{y}}$ might represent a high quality vector of scores which serves as a surrogate for the relevance vector, $\mathbf{r}$. The greater the correlation with this high quality surrogate, the better the retrieval. If we treat the vector $\tilde{\mathbf{y}}$ as a high quality surrogate, then we can replace it with a set of scores which we know to be very good. For example, the combination of scores from multiple systems often, in general, results in very good retrieval performance [Montague and Aslam, 2001]. We can treat the correlation between the retrieval, $\mathbf{y}$, and the combined scores as a predictor of performance. Assume that we are given m retrievals, $\mathbf{y}_i$, for the same n documents. We will represent the mean of these vectors as,

$$\mathbf{y}_\mu = \frac{1}{m}\sum_{i=1}^{m}\mathbf{y}_i \tag{4.8}$$

We use the mean vector as an approximation to relevance. Because $\mathbf{y}_\mu$ represents a very good retrieval, we hypothesize that a strong similarity between $\mathbf{y}_\mu$ and $\mathbf{y}$ will correlate positively with system performance. We use Pearson's product-moment correlation to measure the similarity between these vectors,

$$\rho(\mathbf{y}, \mathbf{y}_\mu) = \frac{\mathbf{y}^\mathsf{T}\mathbf{y}_\mu}{\|\mathbf{y}\|_2\|\mathbf{y}_\mu\|_2} \tag{4.9}$$

Note the similarity between Equation 4.7 and 4.9. A form of this type of precision prediction was proposed by Aslam and Pavlu for ranking queries according to difficulty as opposed to retrieval according to performance [Aslam and Pavlu, 2007].

The retrievals contained in the TREC data consist of multiple score vectors for each query. Therefore, the data from our generalizability experiments allows us to measure the effect of replacing $\tilde{\mathbf{y}}$ with $\mathbf{y}_\mu$. We present the results in Table 4.4. In almost every collection, a retrieval's similarity to the combined scores, $\mathbf{y}_\mu$, is more highly-correlated with performance.

We believe that autocorrelation is, like multiple-retrieval algorithms, approximating a good ranking; in this case by diffusing scores. However, if $\tilde{\mathbf{y}}$ is a reasonable surrogate, then score diffusion tends to, in general, improve performance. Our results demonstrate that this approximation is not as powerful as information from multiple retrievals. Nevertheless, in situations where this extra information is lacking, perhaps we can develop techniques to use information from topically-related documents to systematically improve retrieval scores in a system-agnostic manner.

4.6 Summary

In this chapter, we demonstrated a correlation between retrieval performance and local score consistency. This correlation is comparable with other performance predictors in the

	I_M	$\rho(\mathbf{y}, \mathbf{y}_\mu)$
trec3	**0.461**	0.439
trec4	0.396	**0.482**
trec5	0.277	**0.459**
trec6	0.227	**0.428**
trec7	0.326	**0.430**
trec8	0.396	**0.508**
robust03	0.354	**0.385**
robust04	0.308	**0.384**
robust05	0.249	**0.377**
terabyte04	0.245	**0.420**
terabyte05	0.306	**0.434**
trec4-spanish	0.388	**0.398**
trec5-spanish	0.458	**0.484**
trec5-chinese	0.199	**0.379**
trec6-chinese	0.276	**0.353**
ent05	0.181	**0.305**

Table 4.4. Using a higher quality surrogate. We compare predictiveness of autocorrelation to that of the correlation of $\mathbf{y}$ with interpolated scores from alternate retrievals. We consider the interpolation vector $\mathbf{y}_\mu$ to be a high quality surrogate for relevance.

literature. Local consistency exhibits this correlation across a diverse set of retrieval methods and corpora. We believe that one of the explanations for this correlation is the systemic absence of local consistency as a design principle. Based on an informal analysis, we also believe that our predictor suggests a possible solution this lack of local consistency. In the next part of the thesis, we will develop this idea into a robust and general solution to local score inconsistency.

PART II

REGULARIZATION OF RETRIEVAL SCORES

CHAPTER 5

LOCAL SCORE REGULARIZATION

There is a correlation between local score consistency and retrieval performance. But a correlation alone only suggests exploring local consistency as a system design principle. In this chapter, we propose the following causal hypothesis,

Hypothesis 5.1. *Given a set of retrieval scores, increasing the local consistency of the scores improves retrieval performance.*

We will test this hypothesis by defining an optimization problem whose objective function maximizes local consistency.[1]

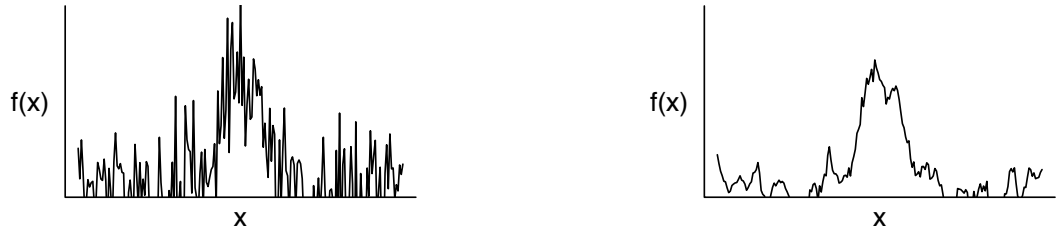
We treat a retrieval as a mapping or function from documents in the collection to a real value. For example, all of the algorithms from Chapter 2 provide a score for each document. Mathematically, given a query, q, a set retrieval model provides a function, $f_q : \mathcal{D} \rightarrow \Re$, from documents to scores; we refer to f_q as the *initial score function* for a particular query. The argument of this function is the retrieval system's representation of a document. The values of the function induce a ranking. Notice that we index functions by the query. We do this to emphasize the fact that, in information retrieval, the score function over all documents will be different for each query. Although we drop the index for notational convenience, the reader should keep in mind that this is a function for a particular query. In this chapter, we will examine the behavior of score functions for ranked retrieval models with respect to the geometry of the underlying domain, $\mathcal{D}$.

One way to describe a function, regardless of its domain, is by its *smoothness*. The smoothness of a function might be measured, for example, by its continuity, as in Lipschitz continuity. In many situations, we prefer functions which exhibit higher smoothness. For example, consider the one-dimensional functions in Figure 5.1. If we assume that local consistency or continuity in the function is desirable, then the function depicted in the Figure 5.1b is preferable because it is smoother.

If only presented with the function in Figure 5.1a, we can procedurally modify the function to better satisfy our preference for smooth functions. The result may be the function in Figure 5.1b. Post-processing a function is one way to perform *regularization* [Chen and Haykin, 2002]. In our work, we regularize initial score functions. Because our analysis and regularization is local to the highest scored documents, we refer to this process as *local score regularization*.

When our domain was the real line, we wanted the value of the function at two points, $f(x_1)$ and $f(x_2)$, to be similar if the distance between the two points, $|x_1 - x_2|$, was small. In

[1] Baliński and Daniłowicz recently proposed a similar score-based objective [Baliński and Daniłowicz, 2005]. Though a solution is presented, we are not aware of any experimental results.



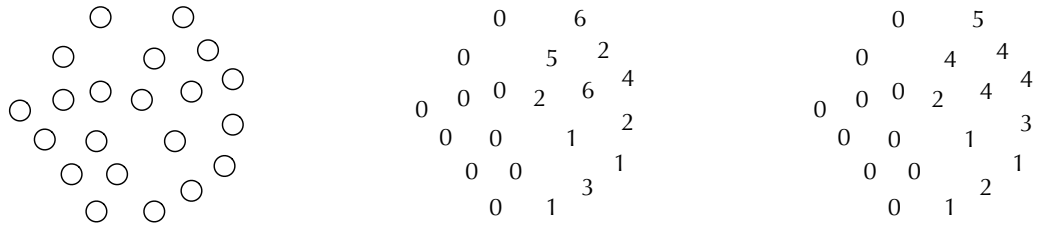
(a) Unregularized (b) Regularized

Figure 5.1. Functions in one dimension. Each value on the horizontal axis may, for example, represent a one-dimensional classification code such as a linear library ordering of books. The functions in these figures assign a value to each point on the real line and may represent relevance. If a set of functions are intended to describe the same phenomenon or signal, we can develop criteria for preferring one function over another. If we prefer smoother function, we would dismiss the function in a in favor of the function in b. The process of smoothing the function in a into the function in b is a type of regularization.

information retrieval, our domain is the set of documents and we want the value of the function for two documents to be similar if the "distance between two documents" is small. We adopt a topic-based distance and consider two documents close if they are topically-related. We will refer to this topical relationship as topical *affinity*. Affinity between documents can be measured using techniques from Section 2.3. We would like two documents which share the same topic to receive similar scores. We depict this graphically in Figure 5.2a for documents in a two-dimensional embedding space. When presented with a query, the retrieval system computes scores for each document in this space (Figure 5.2b); this is our initial score function. We regularize a function into order to improve the consistency of scores between neighboring documents. This is depicted graphically in Figure 5.2c where the value of the function is smoother in the document space. Of course, realistic collections often cannot be visualized like this two-dimensional example. Nevertheless, the fundamental regularization process remains roughly the same.

5.1 Problem Statement

We now formally define the test of Hypothesis 5.1. The **input** is a vector of document scores. Although the system usually scores all n documents in the collection, we consider only the top $\tilde{n}$ scores. The $\tilde{n} \times 1$ vector, $\mathbf{y}$, represents these scores. This vector may be normalized if desired. For example, we normalize this vector to have zero-mean and unit variance. The **output** is the vector of regularized scores represented by the $\tilde{n} \times 1$ vector $\mathbf{f}$. The objective of the regularization process is to improve the local consistency of the scores. If the ranking induced by $\mathbf{f}$ results in performance superior to the ranking induced by $\mathbf{y}$, then we claim to have evidence for Hypothesis 5.1. We will measure performance using



(a) Collection (b) Scored Collection (c) Regularized

Figure 5.2. Regularizing retrieval scores. Documents in a collection can often be embedded in a vector space as shown in a. When presented with a query, a retrieval system provides scores for all of the documents in the collection b. Score regularization refers to the process of smoothing out the retrieval function such that neighboring documents receive similar scores (c).

mean average precision which provides a standard and stable evaluation metric [Buckley and Voorhees, 2000].

5.2 Local Score Regularization[2]

Given the initial scores as a vector, $\mathbf{y}$, we would like to compute a set of regularized scores, $\mathbf{f}$, for these same documents. To accomplish this, we use two contending objectives: score consistency between related documents and score consistency with the initial retrieval. These two objectives are depicted graphically for a one-dimensional function in Figure 5.3. Let $\mathcal{S}(\mathbf{f})$ be a cost function associated with the inter-document consistency of the scores, $\mathbf{f}$; if related documents have very inconsistent scores, then the value of this function will be high. Let $\mathcal{E}(\mathbf{f}, \mathbf{y})$ be a cost function measuring the consistency with the original scores; if documents have scores very inconsistent with their original scores, then the value of this function will be high. For mathematical simplicity, we use a linear combination of these objectives for our composite objective function,

$$\mathcal{Q}(\mathbf{f}, \mathbf{y}) \quad = \quad \mathcal{S}(\mathbf{f}) + \mu\mathcal{E}(\mathbf{f}, \mathbf{y}) \tag{5.1}$$

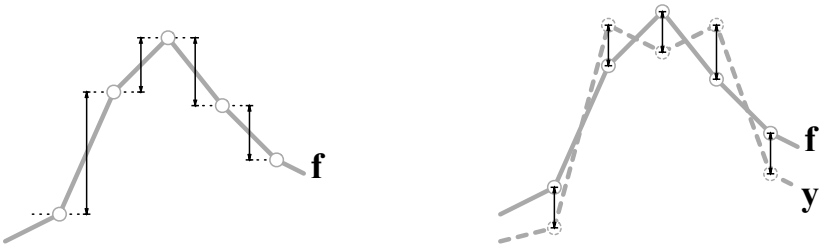
where μ is a parameter allowing us to control how much weight to place on inter-document smoothing versus consistency with the original score.[3]

5.2.1 Measuring Inter-Document Consistency

Inter-document relatedness is represented by the graph, $\mathbf{W}$, defined in Section 2.3.3 where W_{ij} represents the affinity between documents i and j. We define our graph so that there are no self-loops ($W_{ii} = 0$). A set of scores is considered smooth if related documents

[2]We present a regularization method which applies previous results from machine learning [Zhou et al., 2004]. We will review these results in the vocabulary of information retrieval. More thorough derivations can be found in cited publications.

[3]These functions operate on the entire vector $\mathbf{f}$ as opposed to element-wise.



(a) Smoothness Constraint (b) Error Constraint

Figure 5.3. Smoothness and error constraints for a function on a linear graph. In Figure a, the smoothness constraint penalizes functions where neighboring nodes in **f** receive different values. In Figure b, the error constraint penalizes functions where nodes in **f** receive values different from the corresponding values in **y**.

have similar scores. In order to quantify smoothness, we define the cost function, $\mathcal{S}(\mathbf{f})$, which penalizes inconsistency between related documents,

$$\mathcal{S}(\mathbf{f}) = \sum_{i,j=1}^{\tilde{n}} W_{ij} \left(f_i - f_j\right)^2 \tag{5.2}$$

We measure inconsistency using the weighted difference between scores of neighboring documents.[4]

The constraint in Equation 5.2 bears a close relationship to the Moran autocorrelation in Equation 4.5. We can make the relationship clear by rearranging terms in Equation 5.2,

$$\sum_{i,j=1}^{\tilde{n}} W_{ij} \left(f_i - f_j\right)^2 = 2\sum_{i=1}^{\tilde{n}} f_i^2 d_i - 2 \sum_{i,j=1}^{\tilde{n}} W_{ij} f_i f_j \tag{5.3}$$

where $d_i = \sum_{j=1}^{\tilde{n}} W_{ij}$. The first term in the constraint provides a weighted L_2 regularization of the solution while the second term penalizes solutions with low autocorrelation.

In spectral graph theory, Equation 5.2 is known as the Dirichlet sum [Chung, 1997]. We can rewrite the Dirichlet sum in matrix notation,

$$\sum_{i,j=1}^{\tilde{n}} W_{ij} \left(f_i - f_j\right)^2 \;\; = \;\; \mathbf{f}^{\mathsf{T}}(\mathbf{D} - \mathbf{W})\mathbf{f} \tag{5.4}$$

where $\mathbf{D}$ is the diagonal matrix defined as $D_{ii} = d_i$. The matrix $(\mathbf{D} - \mathbf{W})$ is known as the *combinatorial Laplacian* which we represent by Δ_C. The graph Laplacian can be viewed as the discrete analog of the Laplace-Beltrami operator. Because the Laplacian can be used to compute the smoothness of a function, we may abstract Δ_C and replace it with alternative

[4]The local, discrete Lipschitz constant for a document, i, can be thought of as $\max_j \left(W_{ij} \| f_i - f_j \|\right)$. Although similar, the local Lipschitz measure is much less forgiving to discontinuities in a function. Because our retrieval function can be thought of as a very peaked or spiky function due to the paucity of relevant documents, we adopt the Laplacian-based measure.

formulations of the Laplacian which offer alternative measures of smoothness. For example, the *normalized Laplacian* is defined as,

$$\begin{aligned} \Delta_N &= \mathbf{D}^{-1/2}\Delta_C\mathbf{D}^{-1/2} \\ &= \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \end{aligned} \tag{5.5}$$

measures the degree-normalized smoothness as,

$$\mathbf{f}^{\mathsf{T}}\Delta_N\mathbf{f} = \sum_{i,j=1}^{\tilde{n}} \frac{W_{ij}}{D_{ii}D_{jj}} \left(f_i - f_j\right)^2 \tag{5.6}$$

The *approximate Laplace-Beltrami operator* is a variation of the normalized Laplacian which uses a modified affinity matrix [Lafon, 2004]. The approximate Laplace-Beltrami operator is defined as,

$$\Delta_A = \mathbf{I} - \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{W}}\hat{\mathbf{D}}^{-1/2} \tag{5.7}$$

where we use the adjusted affinity matrix $\hat{\mathbf{W}} = \mathbf{D}^{-1}\mathbf{W}\mathbf{D}^{-1}$ with $\hat{D}_{ii} = \sum_{j=1}^{\tilde{n}} \hat{W}_{ij}$. The approximate Laplace-Beltrami operator theoretically addresses violations of the uniform sampling assumption. Because the graph $\mathbf{W}$ will be built from a biased sample, we adopt the approximate Laplace-Beltrami operator (Equation 5.7) in our work. We examine the effect of this choice on the regularization performance in Section 5.4.1.

In Section 2.3.3, we argued that visualization based on the eigenvectors of the Laplacian was not suitable for visualization because it ignored subtle aspects of the graph. The use of the Laplacian in this section, therefore, may seem ill-founded. However, we point out that it is only the embedding process—that is, taking the bottom two eigenvectors or, equivalently, the low frequency harmonics—which makes the visualization globally biased. In this section, we do not perform any eigendecomposition and therefore the Laplacian captures local as well as global behavior.

The value of the objective, $\mathcal{S}(\mathbf{f})$ is small for smooth functions and large for non-smooth function. Unconstrained, however, the function minimizing this objective is the constant function

$$\text{argmin}_{\mathbf{f}}\mathcal{S}(\mathbf{f}) = \mathbf{e}$$

In the next section, we will define a second objective which penalizes regularized scores inordinately inconsistent with the initial retrieval.

5.2.2 Measuring Consistency with Initial Scores

We define a second objective, $\mathcal{E}(\mathbf{f}, \mathbf{y})$, which penalizes inconsistencies between the initial retrieval scores, $\mathbf{y}$, and the regularized scores, $\mathbf{f}$,

$$\mathcal{E}(\mathbf{f}, \mathbf{y}) = \sum_{i=1}^{\tilde{n}} (f_i - y_i)^2 \tag{5.8}$$

The regularized scores, $\mathbf{f}$, minimizing this function would be completely consistent with the original scores, $\mathbf{y}$; that is, if we only minimize this objective, then the solution is $\mathbf{f} = \mathbf{y}$.

5.2.3 Minimizing the Objective Function

In the previous two sections, we defined two constraints, $\mathcal{S}(\mathbf{f})$ and $\mathcal{E}(\mathbf{f}, \mathbf{y})$, which can be combined as a single objective, $\mathbf{f}$. Formally, we would like to find the optimal set of regularized scores, $\mathbf{f}^*$, such that,

$$\mathbf{f}^* = \text{argmin}_{\mathbf{f} \in \Re^{\tilde{n}}} \mathcal{Q}(\mathbf{f}, \mathbf{y}) \tag{5.9}$$

In this section, we will describe two solutions, one iterative and one closed-form, to compute the regularized scores $\mathbf{f}^*$.

Our iterative solution to this optimization interpolates the score of a document with the scores of its neighbors. Metaphorically, this process, at each iteration, *diffuses* scores on the document graph. This is accomplished mathematically by defining a diffusion operator, $\mathbf{S}$, for each Laplacian.

	$\mathbf{S}$
Δ_C	$\mathbf{W}$
Δ_N	$\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
Δ_A	$\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{W}}\hat{\mathbf{D}}^{-1/2}$

Given this operator, the score diffusion process can be formulated as,

$$\mathbf{f}^{t+1} = (1-\alpha)\mathbf{y} + \alpha\mathbf{S}\mathbf{f}^t \tag{5.10}$$

where $\alpha = \frac{1}{1+\mu}$ [Zhou et al., 2004]. We can initialize the regularized scores such that $\mathbf{f}^0 = \mathbf{y}$. As t approaches ∞, the regularized scores, $\mathbf{f}^t$, converge on the optimal scores, $\mathbf{f}^*$. Because we build our graph using a nearest-neighbor technique, this solution also has close relationship to nonparametric regression [Cover, 1968; Devroye, 1978]. In particular, it is an iterated nearest-neighbor regression in the ambient space. The iterative diffusion in Equation 5.10 provides an intuition for the solution to our optimization.

We can also derive a closed form solution to Equation 5.9. We begin by taking the derivative of $\mathcal{Q}(\mathbf{f}, \mathbf{y})$ with respect to $\mathbf{f}$,

$$\frac{\partial}{\partial \mathbf{f}} \mathcal{Q}(\mathbf{f}, \mathbf{y}) = \Delta\mathbf{f} + \mu(\mathbf{f} - \mathbf{y})$$

Setting this equal to zero,

$$\begin{aligned}
\Delta\mathbf{f}^* + \mu(\mathbf{f}^* - \mathbf{y}) &= 0 \\
\alpha\Delta\mathbf{f}^* + (1-\alpha)\mathbf{f}^* - (1-\alpha)\mathbf{y} &= 0 \\
(\alpha\Delta + (1-\alpha)\mathbf{I})\mathbf{f}^* &= (1-\alpha)\mathbf{y} \\
\mathbf{f}^* &= (1-\alpha)(\alpha\Delta + (1-\alpha)\mathbf{I})^{-1}\mathbf{y}
\end{aligned} \tag{5.11}$$

where α is defined above. In our work, we will be using this closed form solution.

Our final score regularization algorithm is presented in Figure 5.4. Note that the affinity matrix computed in Step 1 is used for adding elements to $\mathbf{W}$ in Step 2 and does not define $\mathbf{W}$ itself unless $k = \tilde{n}$. We depict a graph with unregularized and regularized scores in Figure 5.5.

Local Score Regularization

1. compute $\tilde{n} \times \tilde{n}$ affinity matrix,$\mathbf{A}$
2. add the k nearest neighbors for each document to $\mathbf{W}$
3. compute Laplacian, Δ
4. $\mathbf{f}^* = (1-\alpha)\left(\alpha\Delta + (1-\alpha)\mathbf{I}\right)^{-1}\mathbf{y}$

$\tilde{n}$	number of document scores to regularize
$\mathbf{y}$	top $\tilde{n}$ initial retrieval scores
k	number of neighbors to consider
α	parameter favoring inter-document consistency
$\mathbf{f}^*$	regularized scores

Figure 5.4. Local Score Regularization Algorithm. Inputs are $\tilde{n}$, $\mathbf{y}$, k, and α. The output is the a length $\tilde{n}$ vector of regularized scores, $\mathbf{f}^*$.

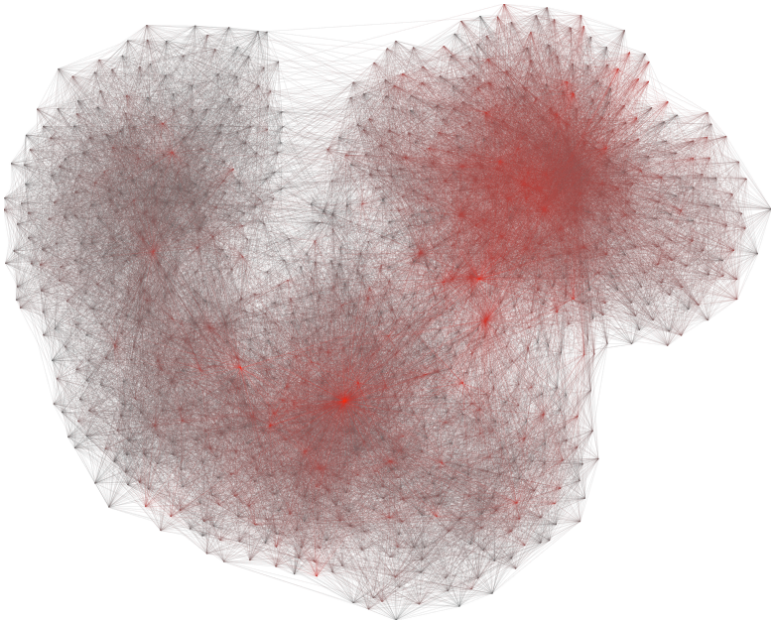
5.3 Experiments

We conducted two sets of experiments. The first set of experiments studies the behavior of regularization in detail for four retrieval algorithms: one vector space model algorithm (Okapi), two language modeling algorithms (query likelihood, relevance models), and one feature-based algorithm (Markov random field); we will abbreviate these okapi, QL, RM, and MRF. We present detailed results demonstrating improvements and parameter stability. We will refer to these as the *detailed experiments*. The second set of experiments applies regularization to automatic runs submitted to the TREC ad hoc retrieval track. These experiments demonstrate the generalizability of regularization. A detailed description of these initial retrievals can be found in Appendix A.

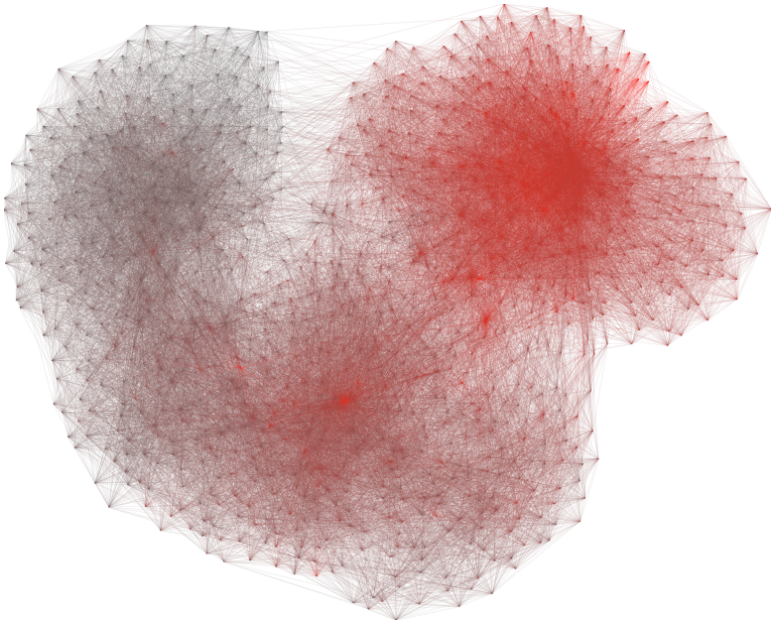
The regularization parameters consist of the degree of regularization, α, and the number of neighbors used for defining the graph, k. In the detailed experiments, we used cosine similarity for the okapi baseline and the diffusion kernel for the QL, RM, and MRF baselines. When our similarity measure is the diffusion kernel, we also train the bandwidth parameter, t. Parameter values considered are,

parameter	range
α	[0.1-0.9; 0.1]
k	{5, 10, 25}
t	{0.1, 0.25, 0.50, 0.75, 0.90}

We describe the data for our generalizability experiments in Appendix A. Due to the large number of runs, we fix $k = 25$ and sweep α between 0.05 and 0.95 with a step size of 0.05. The optimal α is selected using 10-fold cross validation optimizing mean average precision.



(a) unregularized scores



(b) regularized scores

Figure 5.5. Unregularized and regularized scores for the query "U. S. Restaurants in Foreign Lands".

We normalized all scores to zero mean and unit variance for empirical and theoretical reasons [Belkin et al., 2004; Montague and Aslam, 2001].

5.4 Results

5.4.1 Detailed Experiments

Our first set of experiments explored the impact of score regularization on four state-of-the-art baselines. We present precision-recall curves for regularizing these scores in Figures 5.6 (trec12) and 5.7 (robust). The detailed tables of these results showing regularization for different values of $\tilde{n}$ can be found in Appendix D.

We notice that mean average precision improves for all baseline algorithms. These improvements are all significant with $p < 0.05$ using the Wilcoxon test. The gains for query likelihood and the Markov random field model are all approximately 10% relative to the baseline. Regularizing okapi scores results in a smaller relative gain (6-9%). We find weakest improvements ($\tilde{3}$%) with our strongest baseline, relevance models. Nevertheless, even this run sees significant gains in mean average precision.

The precision-recall graphs in Figures 5.6 and 5.7 can be used to detect the location of improvements with respect to the ranked list. All of the improvements from regularization affect the middle-recall parts of ranked lists, resulting in a ballooning out of the precision-recall curve between recall points of 0.20 and 0.60. At low recall points, we only see slight degradations in performance if any. The improvements resulting from regularization, therefore, do not indicate that we are trading high precision for improvements in mean average precision; the gains are consistent across all recall points.

In order to examine the performance changes contributing to the mean changes, we plot the distribution of relative changes in Figure 5.8. The red bars represent improvements; blue bars represent degradations. In all cases, we expect there to be more improvements than degradations. We are interested in measuring the robustness of a net improvement by inspecting the distribution of per-query improvements. A net improvement is unstable if we see many queries substantially improved and many queries substantially degraded. A net improvement is stable, if we see improvements in general and only slight degradations. Across all systems and collections, the improvements are dominated by slight improvements between 0 and 25%. On the other hand, the majority of degradations are also small. With our strongest baseline, these slight changes in performance account for the vast majority of changes in performance, implying that regularization can be applied without fear of significantly hurting some queries. Our other baselines demonstrate many improvements above 25% while avoiding a large number of substantial degradations.

Next, we examine the impact of our choice of Laplacian. In Section 5.2.1, we described three alternative definitions of the graph Laplacian. Because our top $\tilde{n}$ documents were likely to be a non-uniform sample across topics, we adopted the approximate Laplace-Beltrami operator which addresses sampling violations. In order to evaluate this choice of Laplacian, we compared the absolute improvements in performance for all three Laplacians. Our hypothesis was that the approximate Laplace-Beltrami operator, because it is designed to be robust to sampling violations, would result in strong improvements in performance. The results of this comparison are presented in Figure 5.9. In all cases the simple

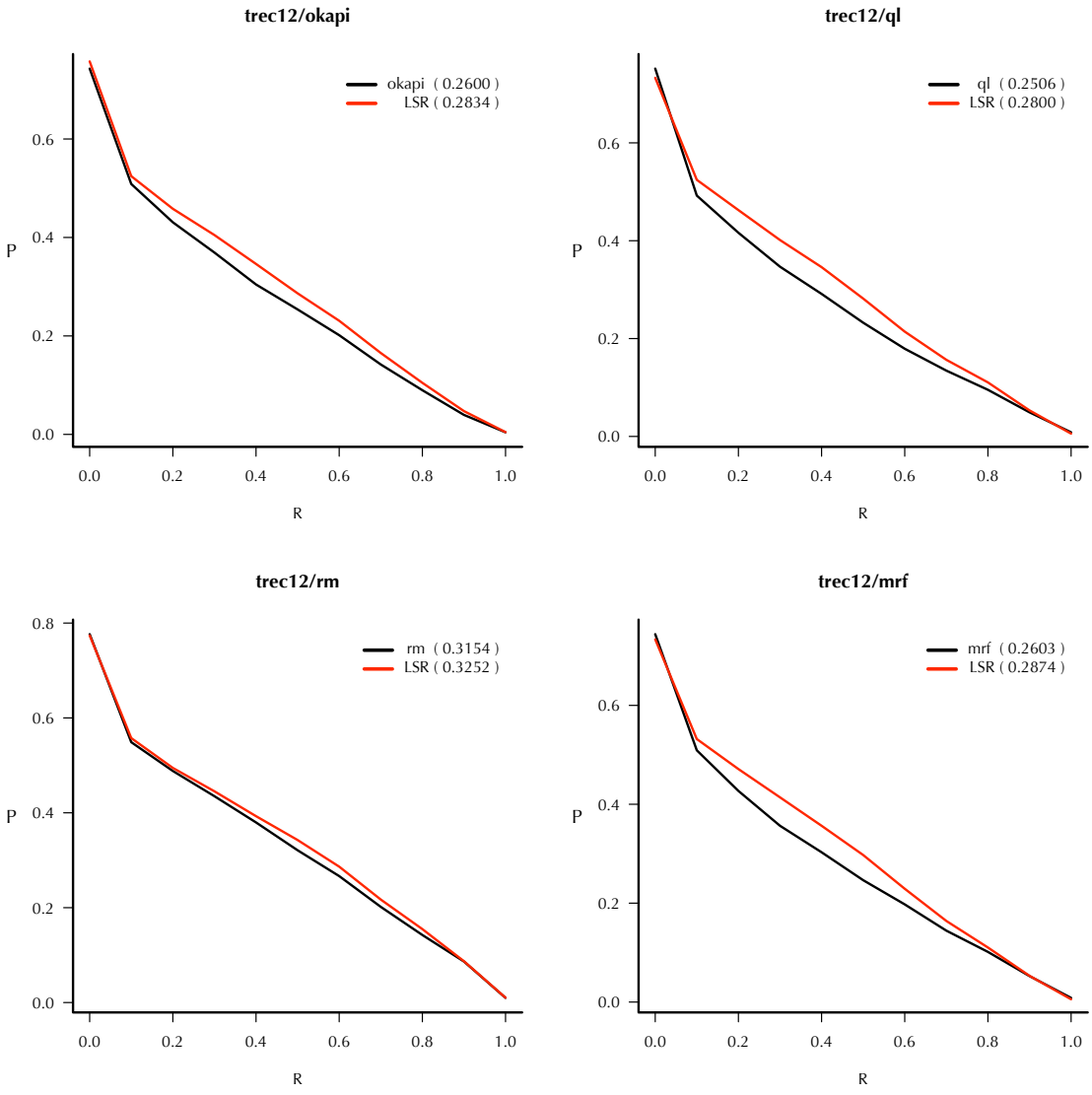


Figure 5.6. Precision-recall curves for regularized trec12 scores. Mean average precision shown in parentheses.

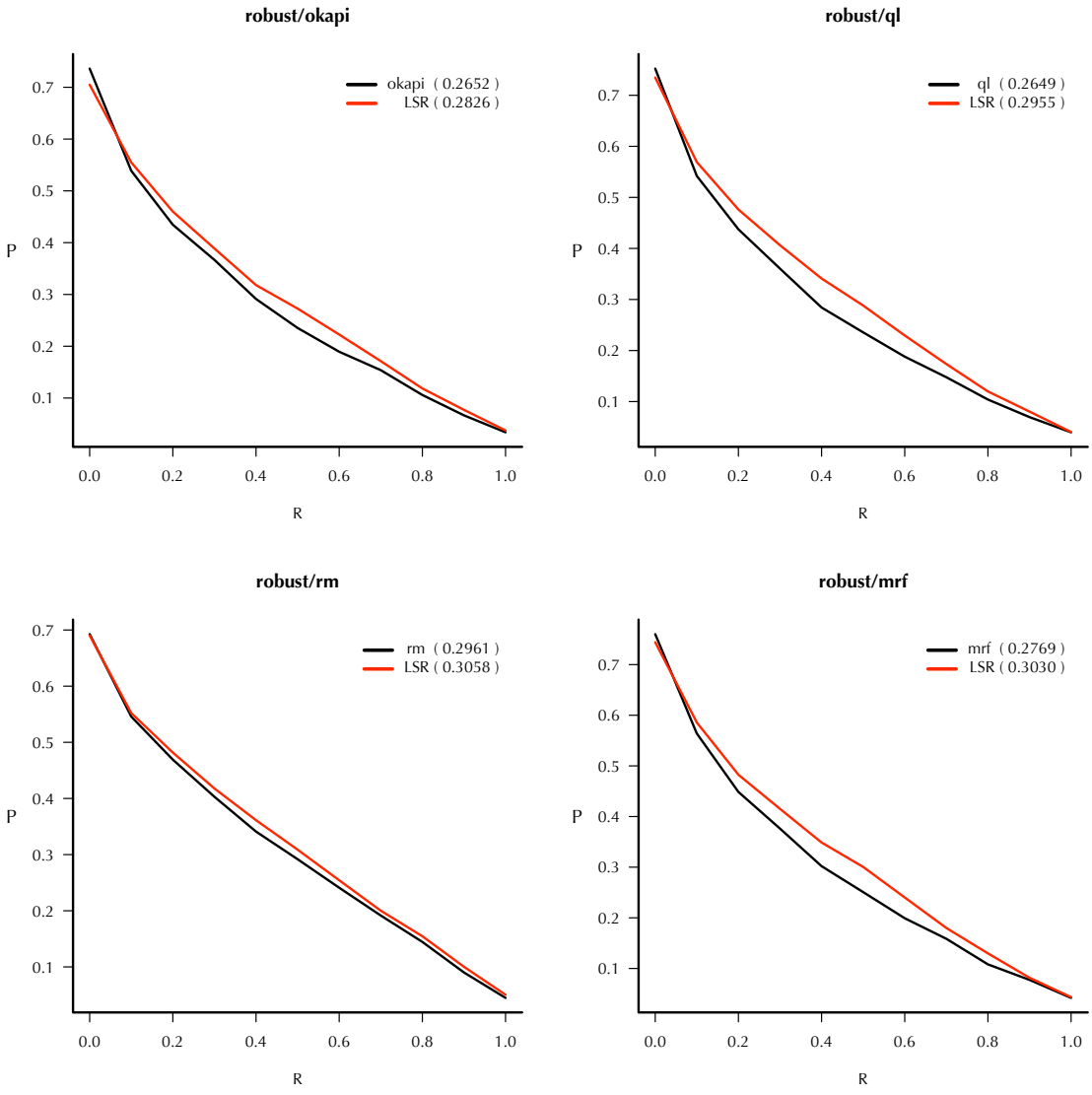


Figure 5.7. Precision-recall curves for regularized robust scores. Mean average precision shown in parentheses.

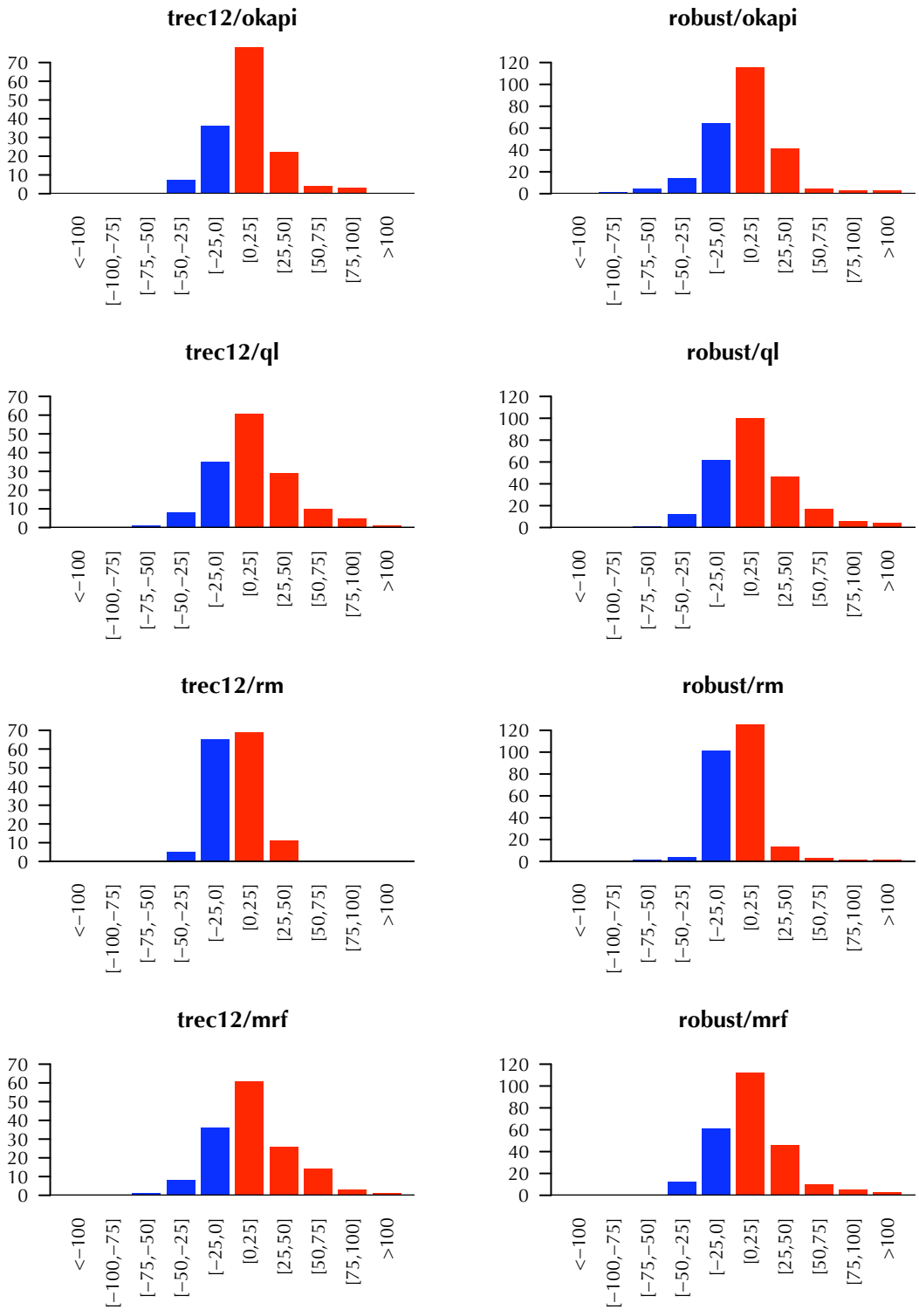


Figure 5.8. Distribution of relative improvements and degradations in performance for detailed experiments.

combinatorial Laplacian clearly underperforms other Laplacians. Recall from Equation 5.2 that, although it weighs the comparisons in scores between documents using W_{ij}, the combinatorial Laplacian does not normalize this weight by the node degrees (ie, D_{ii}). Both the normalized Laplacian (Equation 5.6) and the approximate Laplace-Beltrami operator (Equation 5.7) normalize this weight. However, there do not appear to be significant, consistent advantages to using the approximate Laplace-Beltrami operator over the normalized Laplacian. This result suggests that, while degree normalize is important, our data may not exhibit the appropriate characteristics to notice any benefit to using the approximate Laplace-Beltrami operator.

Our first set of experiments, described in Figures 5.6 and 5.7, demonstrated improvements across all four baseline algorithms. The α parameter controls the degree of regularization and therefore the amount of local consistency introduced. In Figure 5.10, we plot the effect of regularization as a function of this parameter. We see that, in some cases, regularization actually hurts performance at high values of α. This results from the fact that, as α approaches 1, the document scores grow increasingly similar, at some point becoming constant. This effect is more noticeable with our pseudo-relevance baseline, indicating that perhaps the scores may already locally-consistent. We will return to this in observation in the next chapter. Other baselines, however, see improvements from higher ranges of α. Appealing to the diffusion metaphor, a higher α that documents are gaining more information from their neighbors than from their original scores.

One of the core assumptions behind our technique is the presence of a lower-dimensional structure recovered by the graph. The number of neighbors, k, represents how much we trust the ambient affinity measure for this set of documents. If performance improves as we consider more neighbors, graph-based methods are less-justified. In Figure 5.11, we evaluate performance as a function of the number of neighbors. Across all algorithms and all distance measures, we notice a degradation in performance as more neighbors are considered. This occurs even in the presence of a soft nearest neighbor measure such as the diffusion kernel. This behavior might result from several causes. For example, our similarity measure may be inaccurate for larger dissimilarities, resulting in an ability to accurately order several dissimilar documents. Alternatively, we may be experiencing a non-uniform distribution of documents in the ambient space (recall our discussion of this in Section 3.3).

5.4.2 Generalizability Experiments

Our detailed experiments demonstrated the improvement of performance achieved by regularizing three four baselines. We were also interested in the performance over a wide variety of initial retrieval algorithms. We present results for regularizing the TREC submissions in Figures 5.12 and 5.13. Although regularization on average produces improvements, there are a handful of runs for which performance is degraded. Inspecting these baselines of the more dramatic degradations (trec8), we noticed that the scores for these runs had odd distributions, greatly affecting our normalization procedures. Other reductions in performance may be the result of an unoptimized k parameter. Improvements are consistent across collections and languages.

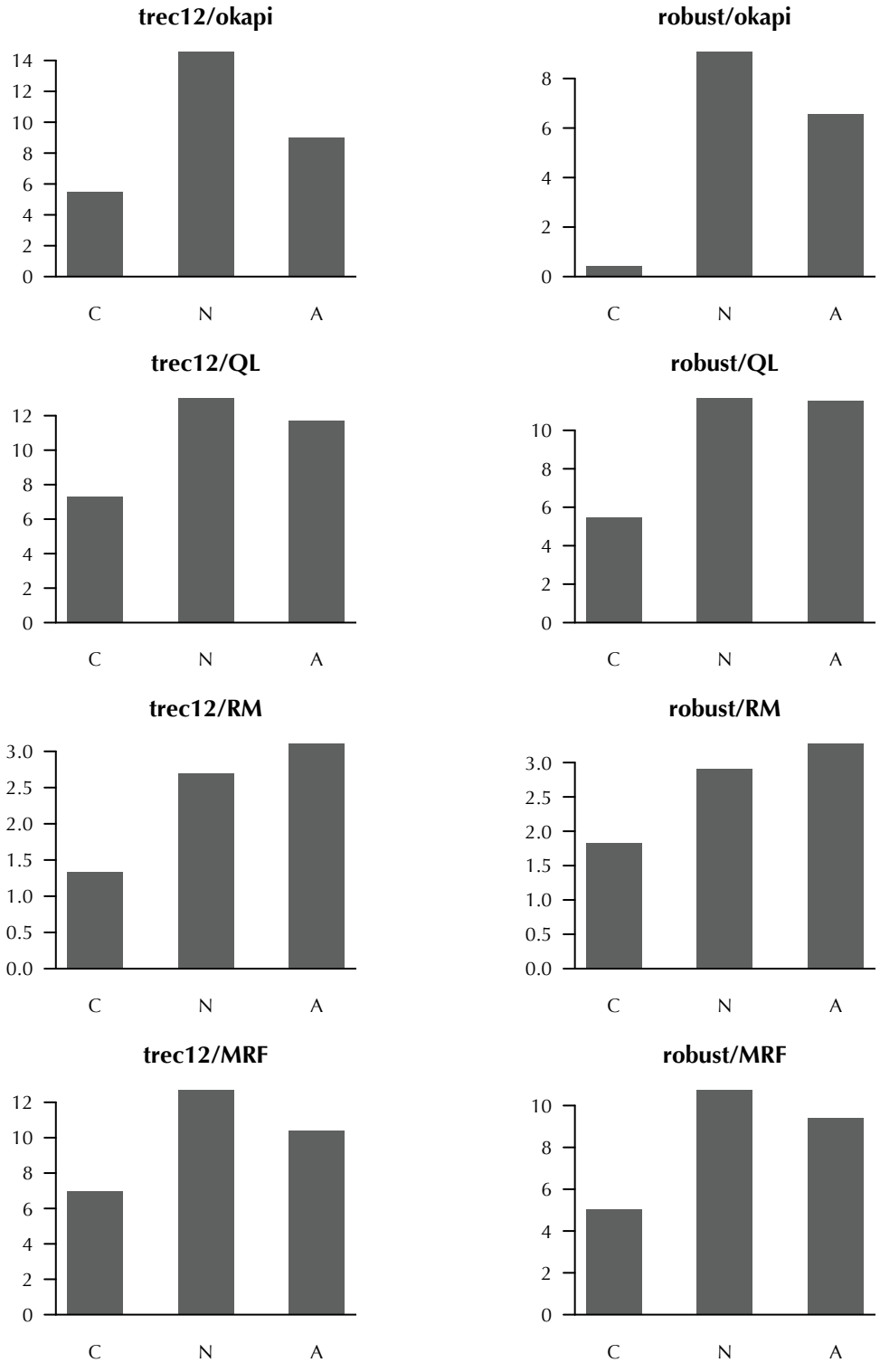


Figure 5.9. Performance improvement as a function of Laplacian type. For each Laplacian described in Section 5.2.1, we maximized mean average precision using 10-fold cross-validation (left: combinatorial Laplacian, center: normalized Laplacian, right: approximate Laplace-Beltrami). The different Laplacians represent different degree normalization techniques.

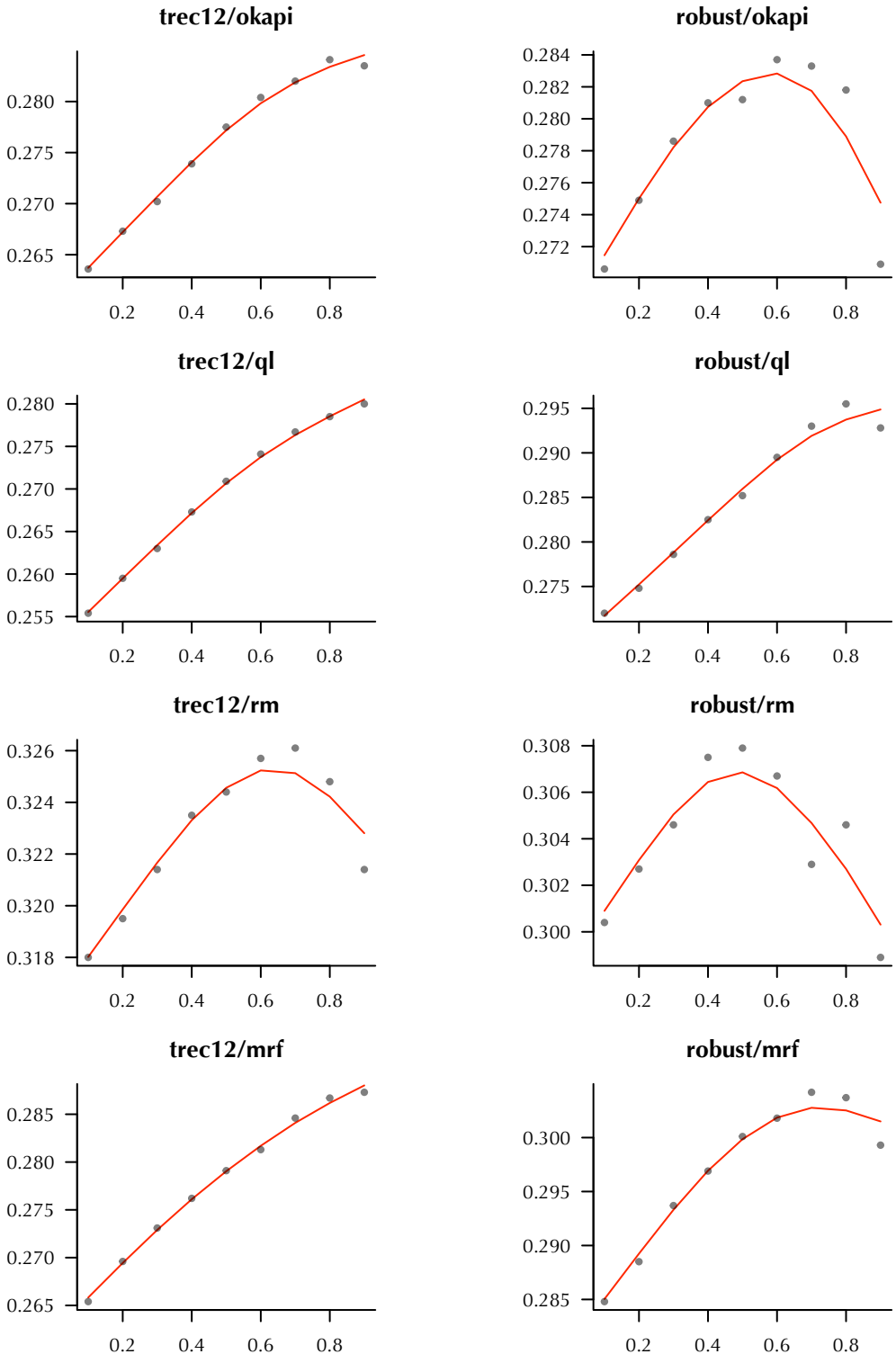


Figure 5.10. Performance as a function of amount of regularization. For each value of α, we selected the values for k and t maximizing mean average precision. A higher value for α results in more aggressive regularization. A low value of α recovers the original scores.

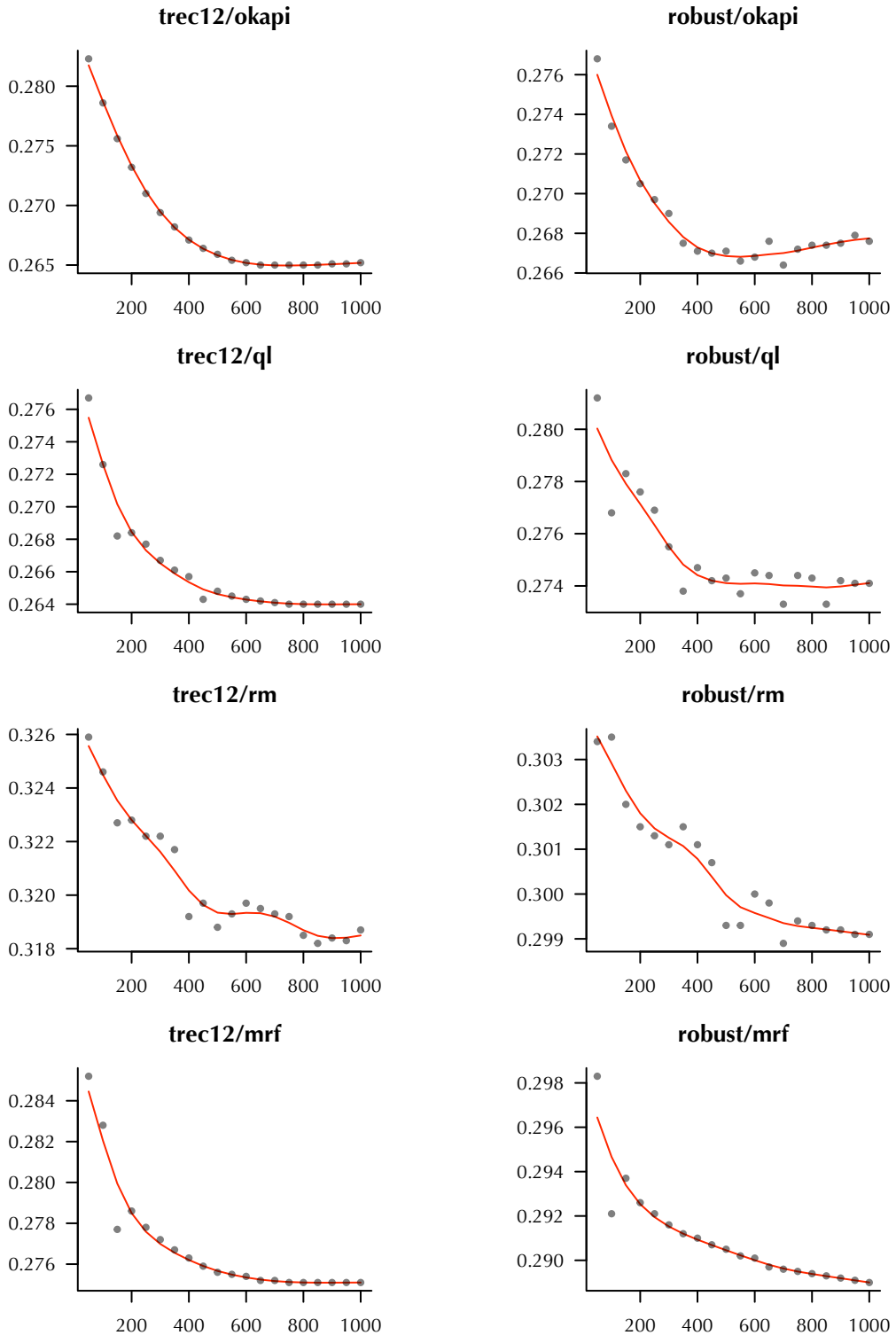


Figure 5.11. Performance as a function of number of neighbors. For each value of k, we selected the value for α and t maximizing mean average precision. If we trust the distance metric, we would expect the performance to increase as we increase the number of neighbors.

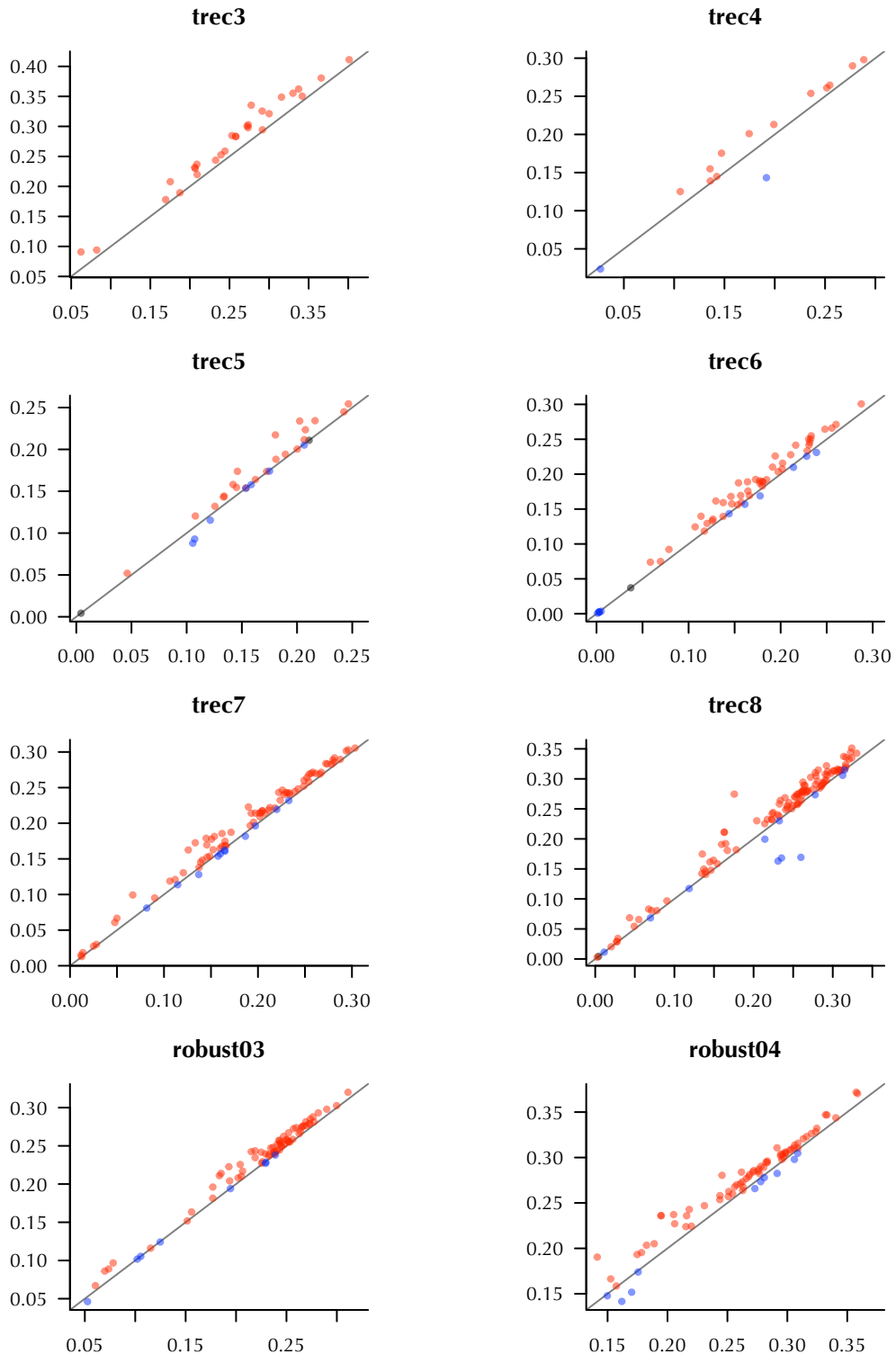


Figure 5.12. Improvement in mean average precision for TREC query-based retrieval tracks. Each point represents a competing run. The horizontal axis indicates the original mean average precision for this run. The vertical axis indicates the mean average precision of the regularization run. Red points indicate an improvement; blue points indicate degradations.

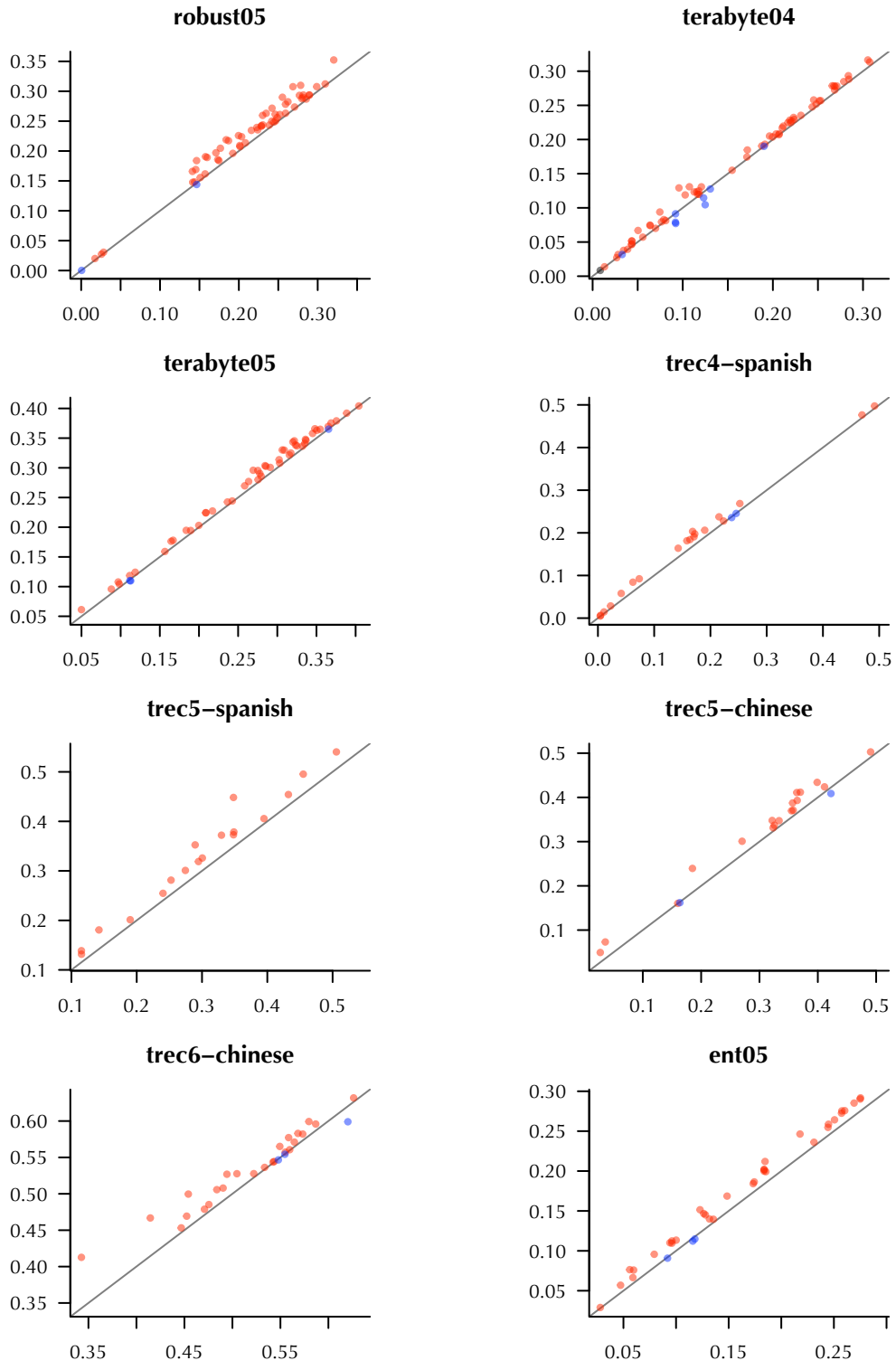


Figure 5.13. Improvement in mean average precision for TREC query-based retrieval tracks. Each point represents a competing run. The horizontal axis indicates the original mean average precision for this run. The vertical axis indicates the mean average precision of the regularization run. Red points indicate an improvement; blue points indicate degradations.

5.5 Discussion

We introduced score regularization in order to test Hypothesis 5.1. The results from our experiments indicate that increasing the local consistency of scores does improve performance.

We have also developed a generic post-processing procedure for improving the performance of arbitrary score functions. The results in Figures 5.12 and 5.13 provide evidence that existing retrieval algorithms can benefit from regularization.

The results in Figures 5.9 and 5.11 suggest that the construction of the diffusion operator is sometimes important for regularization efficacy. Since there are a variety of methods for constructing affinity and diffusion geometries, we believe that this should inspire a formal investigation and comparison of various approaches.

The results in Figure 5.11 also allow us to test the manifold properties of the initial retrieval. The relatively small range of change of the curves for the relevance model run implies that the ambient measure behaves well for the documents in this retrieval. Poorer-performing algorithms, by definition, have a mix of relevant and non-relevant documents. Including more edges in the graph by increasing the value of k will be more likely to relate relevant and non-relevant documents. From the perspective of graph-based methods, the initial retrieval for poorer-performing algorithms should be aggressively sparsified with low values for k. On the other hand, better performing algorithms may benefit less from a graph-based representation allowing us to let k grow. From a geometric perspective, documents from poorer-performing algorithms are retrieved from regions of the embedding space so disparate that topical relationships are poorly-approximated by the ambient affinity. Documents from better performing queries all exist in a region of the embedding space where affinity is well-approximated by the ambient affinity.

We have noted that the aggressiveness of regularization (α) is related to the performance of the initial retrieval. Figure 5.10 demonstrates that smaller values for α are more suitable for better-performing algorithms. This indicates that the use of techniques from precision prediction may help to automatically adjust the α parameter [Carmel et al., 2006; Cronen-Townsend et al., 2002; Yom-Tov et al., 2005].

We mentioned in Chapter 3 that the cluster hypothesis motivates retrieval methods such as cluster-based retrieval. It is worth comparing the effectiveness improvements resulting from regularization to those resulting from other clustering methods. Therefore, we regularized the scores of a strong baseline for a collection used in previous cluster-based retrieval work [Liu and Croft, 2004, 2006]. Results are presented in Table 5.1. The small set of results in this table indicate that the improvements achieved by regularization are as strong as those achieved by the various cluster-based retrieval methods. We speculate that this is because the clustering done by these previous methods considers larger scale than the local methods implicit in regularization. Clustering into large clusters results in introducing relationships between otherwise dissimilar documents and smoothing into more general topics than the query requires.

Finally, we should address the question of efficiency. There are two points of computational overhead in our algorithm. First, the construction of the $\tilde{n} \times \tilde{n}$ affinity matrix requires $O(\tilde{n}^2)$ comparisons. For $\tilde{n} = 1000$, this took approximately 8 seconds. Although most of our experiments use $\tilde{n} = 1000$, we can inspect the improvement in performance

	QL	single link	average link	Ward	CBDM	LSR
AP	0.2179	0.2153	0.2161	0.2160	0.2326	0.2562
WSJ	0.2958	0.2911	0.2902	0.2963	0.3006	0.3141

Table 5.1. Comparison of cluster-based retrieval methods and regularization. Mean average precision is presented for the Associated Press and Wall Street Journal collections [Liu and Croft, 2004]. The columns labeled "single link", "average link", and "Ward" refer to agglomerative clustering methods. The column labeled "CBDM" refers to a cluster-based document expansion method. All non-regularization performance values are copied directly from previous publications. Regularization used a query likelihood baseline which was comparable in terms of performance to the baseline used in the referenced publications.

as a function of the number of documents being regularized, $\tilde{n}$. In Figure 5.14, we notice that performance improves and then plateaus. These results show that $\tilde{n}$ need not be as large as this to achieve improvements. For example, for $\tilde{n} = 100$, this computation takes less than 0.5 seconds. We should also point out that we can compute the entire collection affinity matrix and store it prior to any retrieval. In Figure 5.11, we showed that only very few neighbors were required to perform well, implying that the storage cost can be $O(nk)$.

The second point of computational overhead is in the inversion of the matrix in Equation 5.11. We show running time as a function of $\tilde{n}$ in Figure 5.15. Note that our experiments, although very expensive when $\tilde{n} = 1000$, can be computationally improved significantly by reducing $\tilde{n}$ to 500 which, according to Figure 5.14, would still boost baseline performance. We could also address the inversion by using the iterative solution. In related work, using a pre-computed similarity matrix and an iterative solution allowed real-time pseudo-relevance feedback [Lavrenko and Allan, 2006].

5.6 Conclusions

We have provided substantial evidence that the introduction of local consistency into a set of retrieval scores improves performance. Our results do not suggest a monotonic relationship since the benefits fell after the some amount of regularization. Although we began this chapter intending to test a hypothesis about the relationship between local consistency and performance, as a byproduct, we have developed a black box method for improving the performance arbitrary retrieval algorithms. Having demonstrated demonstrated the effectiveness of regularization, in the subsequent chapters, we will study regularization in more technical detail.

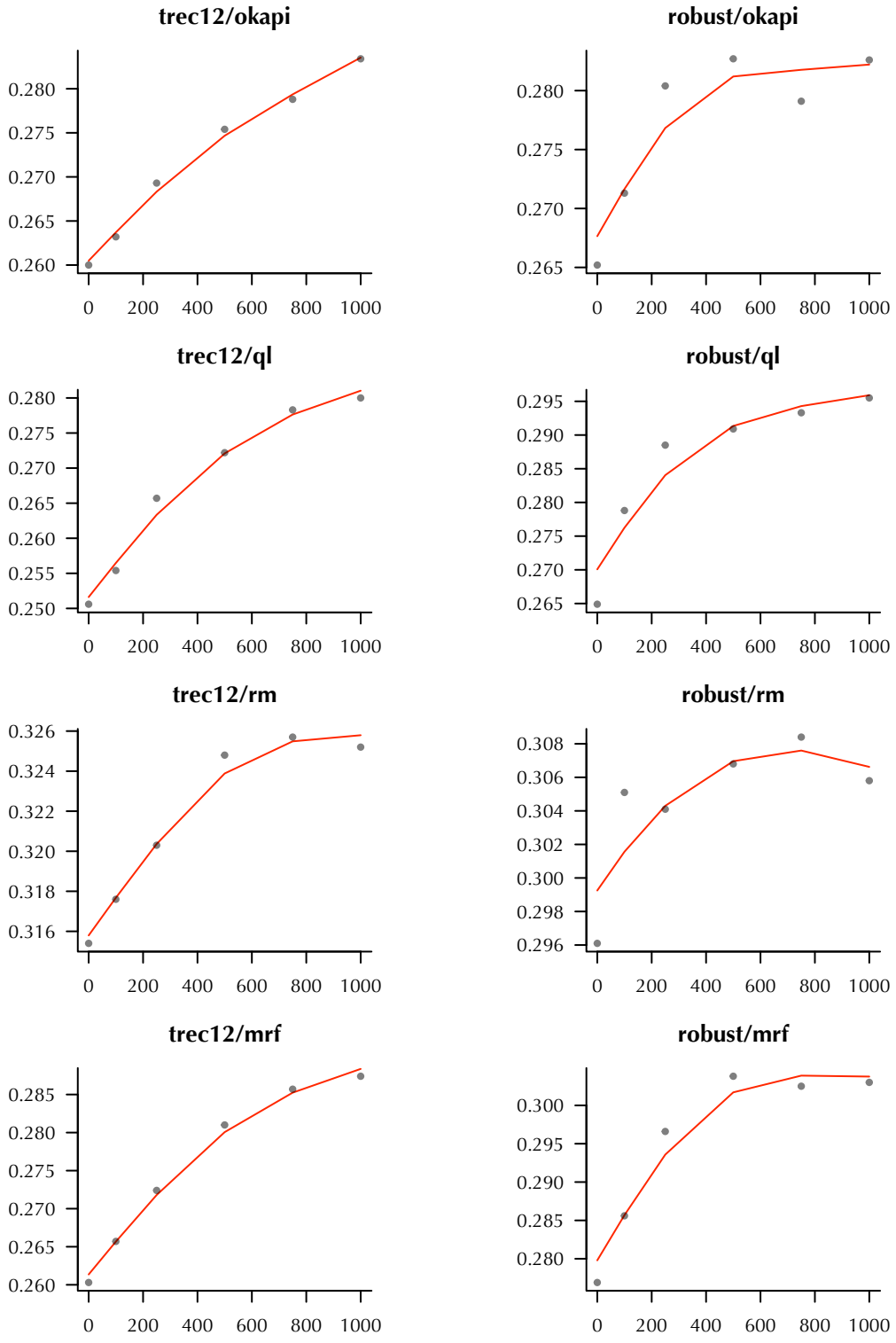


Figure 5.14. Performance as a function of number of documents used for regularization. For each value of $\tilde{n}$, we selected the values for α, k and t maximizing mean average precision. A higher value for $\tilde{n}$ considers more documents in the regularization.

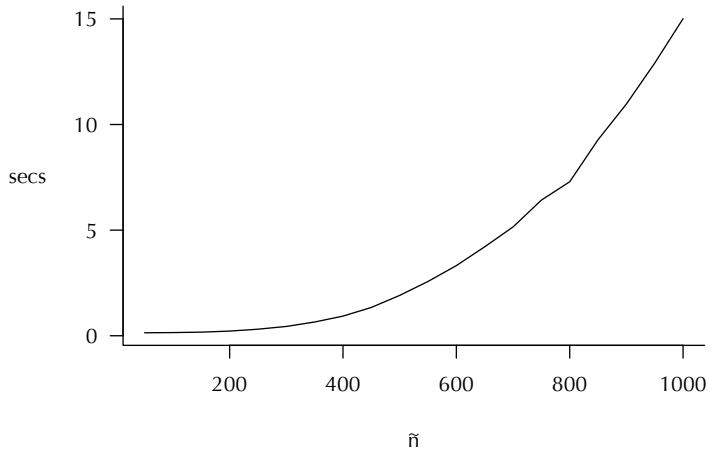


Figure 5.15. Running time as a function of number of documents used for regularization. For each value of $\tilde{n}$, we regularized the scores given a pre-computed affinity matrix.

CHAPTER 6

CONNECTIONS BETWEEN REGULARIZATION AND OTHER RETRIEVAL METHODS

Several classic retrieval methods can be posed as instances of score regularization. We will be focusing on the relationship between these methods and a single iteration of score regularization (Equation 5.10). In previous chapters, we considered only the top $\tilde{n} \ll n$ documents from some initial retrieval. In this section, we may at times consider every document in the collection (ie, $\tilde{n} = n$).

In this chapter, we will demonstrate that regularization is a common criterion for many successful retrieval methods. This approach is similar to Fang *et al*'s study of heuristics shared across formal models of information retrieval [Fang et al., 2004, p. 49; emphasis added],

> Despite the progress in the development of formal retrieval models, good empirical performance rarely comes directly from a theoretically well-motivated model; rather, heuristic modification of a model is often necessary in order to achieve optimal retrieval performance. Indeed, *many empirical studies show that good retrieval performance is closely related to the use of various retrieval heuristics*, especially TF-IDF weighting and document length normalization.

We will show that regularization, like tf.idf and document length normalization, is a property found in many successful retrieval methods.

For each of the methods in this section, we will be asking ourselves the following question: can the final retrieval scores be computed as a function of the initial retrieval scores and a similarity-based adjacency matrix? If the answer to this question is "yes", then we can state that this method is an instance of score regularization.

6.1 Vector Space Model Retrieval

In Section 2.3.1, we represented each document as an L_2 normalized, length-m vector, $\mathbf{d}$. A query can also be represented by a normalized, length-m vector, $\mathbf{q}$. A document's score is the inner product between its vector and the query vector (ie, $y_i = \langle \mathbf{d}_i, \mathbf{q} \rangle$).

Claim 6.1. *Pseudo-relevance feedback in the vector space model using Rocchio is a form of regularization.*

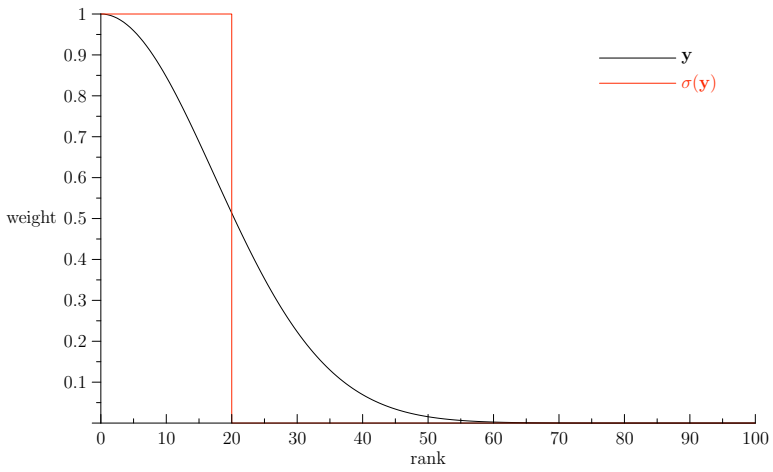


Figure 6.1. Hard weighting function for pseudo-relevance feedback. The horizontal axis represents the documents in decreasing order of $\mathbf{y}$. The function $\sigma(\mathbf{y})$ acts as a filter for pseudo-relevant documents. It sets the score of each of the top r documents to 1.

Proof. First, we note that the similarity between a document and the new query can be written as the combination of the original document score and the sum of similarities to the pseudo-relevant set,

$$\begin{aligned} \langle \mathbf{d}_i, \tilde{\mathbf{q}} \rangle &= \left\langle \mathbf{d}_i, \mathbf{q} + \frac{\alpha}{r} \sum_{j \in R} \mathbf{d}_j \right\rangle \\ &= \langle \mathbf{d}_i, \mathbf{q} \rangle + \frac{\alpha}{r} \left\langle \mathbf{d}_i, \sum_{j \in R} \mathbf{d}_j \right\rangle \\ &= \langle \mathbf{d}_i, \mathbf{q} \rangle + \frac{\alpha}{r} \sum_{j \in R} \langle \mathbf{d}_i, \mathbf{d}_j \rangle \end{aligned} \tag{6.1}$$

Notice here that the first factor in the sum is y_i and the second factor in the sum represents the similarity to the pseudo-relevant documents, $\sum_{j \in R} A_{ij}$. We can rewrite Equation 6.1 in terms of matrix operators to compute the new scores for all documents in the collection. This computation is a function of the initial scores and the inner product affinity matrix,

$$\mathbf{f} = \mathbf{y} + \frac{\alpha}{||\sigma(\mathbf{y})||_1} \mathbf{A} \sigma(\mathbf{y}) \tag{6.2}$$

where $\sigma(\mathbf{y}) : \Re^n \rightarrow \Re^n$ is defined as,

$$\sigma(\mathbf{y})_i = \begin{cases} 1 & \text{if } i \in R \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

We compare $\sigma(\mathbf{y})$ to $\mathbf{y}$ in Figure 6.1. The σ function maps high-ranked documents to pseudo-scores of 1. This behavior replicates the judgment of documents as relevant. From our perspective of score functions, we see that σ acts as a hard filter on the signal $\mathbf{y}$. This demonstrates that Rocchio is an instance of score regularization.

□

Whereas pseudo-relevance feedback incorporates into a query terms from r pseudo-relevant documents, *document expansion* incorporates into a document the terms from its k most similar neighbors [Singhal and Pereira, 1999]. The modified document, $\tilde{\mathbf{d}}$, is defined as,

$$\tilde{\mathbf{d}}_i = \alpha_D \mathbf{d}_i + \frac{1}{k} \sum_{j \in N(i)} \mathbf{d}_j \tag{6.4}$$

where α_D is the weight placed on the original document vector. $N(i)$ is the set of k documents most similar to document i.

Claim 6.2. *Document expansion in the vector space model is a form of regularization.*

Proof. Define the binary matrix $\mathbf{W}$ so that each row i contains k non-zero entries for each of the indices in $N(i)$. We can expand all documents in the collection,

$$\tilde{\mathbf{C}} = \alpha_D \mathbf{C} + \frac{1}{k} \mathbf{W}\mathbf{C} \tag{6.5}$$

Given a query vector, we can score the entire collection,

$$\begin{aligned} \mathbf{f} &= \tilde{\mathbf{C}}\mathbf{q} \\ &= (\alpha_D \mathbf{C} + \frac{1}{k} \mathbf{W}\mathbf{C})\mathbf{q} \\ &= \alpha_D \mathbf{C}\mathbf{q} + \frac{1}{k} \mathbf{W}\mathbf{C}\mathbf{q} \\ &= \alpha_D \mathbf{y} + \frac{1}{k} \mathbf{W}\mathbf{y} \end{aligned} \tag{6.6}$$

The point here is that the score of an expanded document (f_i) is the linear combination of the original score (y_i) and the scores of its k neighbors ($\frac{1}{k} \sum_{j \in N(i)} y_i$). This demonstrates that document expansion is a form of regularization. □

We now turn to the dimensionality reduction school of cluster-based retrieval algorithms. In the previous proof, we expanded the entire collection using the matrix $\mathbf{W}$. Clustering techniques such as Latent Semantic Indexing (LSI) can also be used to expand documents [Deerwester et al., 1990]. LSI-style techniques use two auxiliary matrices: $\mathbf{V}$ is the $k \times n$ matrix embedding documents in the k-dimensional space and $\mathbf{U}$ is $m \times k$ representations of the dimensions in the ambient space. Oftentimes, queries are processed by projecting them into the k-dimensional space (ie, $\tilde{\mathbf{q}} = \mathbf{U}^\mathsf{T}\mathbf{q}$). We use an equivalent formula where we expand documents by their LSI-based dimensions,

$$\tilde{\mathbf{C}} = \lambda \mathbf{C} + (1 - \lambda) \mathbf{V}^\mathsf{T}\mathbf{U}^\mathsf{T}$$

We then score a document by its cluster-expanded representation.[1]

Claim 6.3. *Cluster-based retrieval in the vector space model is a form of regularization.*

[1] In practice, the document representations are only based on the cluster information (ie, $\lambda = 0$). Our ranking function generalizes classic cluster-based retrieval functions.

Proof. Our proof is similar to the proof for document expansion.

$$\begin{aligned}\mathbf{f} &= \tilde{\mathbf{C}}\mathbf{q} \\ &= (\lambda\mathbf{C} + (1-\lambda)\mathbf{V}^\mathsf{T}\mathbf{U}^\mathsf{T})\mathbf{q} \\ &= \lambda\mathbf{y} + (1-\lambda)\mathbf{V}^\mathsf{T}[\mathbf{U}^\mathsf{T}\mathbf{q}] \\ &= \lambda\mathbf{y} + (1-\lambda)\mathbf{V}^\mathsf{T}\mathbf{y}_c \end{aligned} \tag{6.7}$$

Because the dimensions (clusters) are representable in the ambient space, we can score them as we do documents; here, we use the $k \times 1$ vector, $\mathbf{y}_c$ to represent these scores. Essentially, the document scores are interpolated with the scores of the clusters. □

6.2 Language Modeling Retrieval

In the previous section, we demonstrated the equivalence between several vector space model techniques and regularization. In the context of retrieval using language models, we can only show a reduction for pseudo-relevance feedback. This result, though, is significant since it provides an alternative explanation for the success of this method.

Claim 6.4. *Relevance models are a form of regularization.*

Proof. Our proof is based on a similar derivation used in the context of efficient pseudo-relevance feedback [Lavrenko and Allan, 2006]. Recall that we use $(\log \mathbf{C})\tilde{\mathbf{q}}$ to rank the collection. By rearranging some terms, we can view relevance models from a different perspective,

$$\begin{aligned}\mathbf{f} &= (\log \mathbf{C})\tilde{\mathbf{q}} \\ &= (\log \mathbf{C})\left(\lambda\mathbf{q} + \frac{(1-\lambda)}{||\mathbf{y}||_1}\mathbf{C}^\mathsf{T}\mathbf{y}\right) \\ &= \lambda(\log \mathbf{C})\mathbf{q} + \frac{(1-\lambda)}{||\mathbf{y}||_1}(\log \mathbf{C})\mathbf{C}^\mathsf{T}\mathbf{y} \\ &= \lambda\mathbf{y} + \frac{(1-\lambda)}{||\mathbf{y}||_1}\mathbf{A}\mathbf{y} \end{aligned} \tag{6.8}$$

where $\mathbf{A}$ is an $n \times n$ affinity matrix based on inter-document cross-entropy. Since the relevance model scores can be computed as a function of inter-document affinity and the initial scores, this is an instance of score regularization. In fact, iterating the process in Equation 2.13 has been shown to improve performance of relevance models and provides an argument for considering the closed form solution in Equation 5.11 [Kurland et al., 2005].[2] □

Unfortunately, we cannot reduce document expansion in the language modeling framework to regularization. Document expansion in language modeling refers to adjusting the

[2] In Section 2.3.2, we adopted the symmetric diffusion kernel to compare distributions. The cross-entropy measure here is asymmetric and therefore cannot be used in our closed form solution. Nevertheless, our iterative solution is not constrained by the symmetry requirement. Furthermore, theoretical results for Laplacians of directed graphs exist and can be applied in our framework [Chung, 2004; Zhou et al., 2005].

document language models $P(w|\theta_d)$ given information about neighboring documents [Tao et al., 2006]. In this situation, the score function can be written as,

$$\mathbf{f} \;=\; \log\left(\lambda\mathbf{C} + (1-\lambda)\mathbf{AC}\right)\mathbf{q} \tag{6.9}$$

Because the logarithm effectively decouples the document expansion from the document scoring, the approach used in the vector space model proof cannot be used here.

The language modeling approach to cluster-based retrieval is conceptually very similar to document expansion [Liu and Croft, 2004; Wei and Croft, 2006]. The distribution $P(z|D)$ represents the distribution of subtopics or aspects in a document; we also have $P(w|z)$ representing language models for each of our subtopics. When we interpolate these models with the maximum likelihood document models, we get a score function similar to Equation 6.7,

$$\mathbf{f} \;=\; \log\left(\lambda\mathbf{C} + (1-\lambda)\mathbf{V}^{\mathsf{T}}\mathbf{U}^{\mathsf{T}}\right)\mathbf{q} \tag{6.10}$$

where $\mathbf{V}$ is the $k \times n$ distribution $P(z|D)$ and $\mathbf{U}$ is the $m \times k$ distribution $P(w|z)$. Like document expansion scores, the logarithm prevents converting cluster-based expansion into a regularization form.

It is worth devoting some time to Kurland and Lee's cluster-based retrieval model [Kurland and Lee, 2004]. The model is used to perform retrieval in three steps. First, each document is scored according to an expanded document model. Second, an $n \times n$ matrix comparing unexpanded and expanded models is constructed. Finally, each document is scored by the linear interpolation of its original (unexpanded) score and the scores of the nearest expanded documents. To this extent, the model combines regularization and document-expansion retrieval in a language modeling framework. Unfortunately, there do not appear to be experiments demonstrating the effectiveness of each of these steps. Is this model an instance of score regularization? Yes and no. The second interpolation process clearly is an iteration of score regularization. The first score is language model document expansion and therefore not regularization.

Recall that the vector space model allowed fluid mathematical movement from query expansion to regularization to document expansion and finally to cluster-based retrieval. This is not the case for language modeling. Language models have a set of rank-equivalent score functions; we adopt cross entropy in our work. The problem, however, is that measures such as the Kullback-Leibler divergence, cross entropy, and query likelihood all are non-symmetric and therefore not valid inner products. This disrupts the comparison to the vector space model derivations because a smooth transition from regularization (Equation 6.8) to document expansion is impossible.

6.3 Feature-Based Retrieval

Pseudo-relevance feedback in the context of feature-based retrieval can also be reduced to regularization. In the Markov random field model of information retrieval, pseudo-relevance feedback is referred to as *latent concept expansion* (LCE) and has very close theoretical connections to relevance models [Metzler and Croft, 2007]. In simple terms, LCE works by

performing expansion using *expressions* from an initial MRF-based retrieval (Section 2.2.3). More formally, given $P_{G,\Lambda}(D|Q)$, then expansion expressions are weighted according to,

$$\begin{aligned}
P_{H,\Lambda}(e|Q) &= \sum_{D \in \mathcal{R}_Q} \exp\left(F_{DQ}(D,Q) + \log \frac{((1-\alpha)\frac{tf_{e,D}}{|D|} + \alpha\frac{cf_e}{|C|})^{\lambda'_{T_D}}}{\left(\frac{cf_e}{|C|}\right)^{\lambda'_{T_Q}}} \right) \\
&= \sum_{D \in \mathcal{R}_Q} \exp\left(F_{DQ}(D,Q)\right) \frac{\tilde{P}(e|D)^{\lambda'_{T_D}}}{\tilde{P}(e|C)^{\lambda'_{T_Q}}} \\
&= \sum_{D \in \mathcal{R}_Q} P_{G,\Lambda}(D|Q) \frac{\tilde{P}(e|D)^{\lambda'_{T_D}}}{\tilde{P}(e|C)^{\lambda'_{T_Q}}}
\end{aligned}$$

where each expression is weighted by its collection frequency in a manner originally proposed by Li [Li, 2006]. Let $\mathcal{Z}_Q = \sum_{e \in \mathcal{E}} P_{H,\Lambda}(e|Q)$ where $\mathcal{E}$ is the set of features we are selecting from. Metzler and Croft [Metzler and Croft, 2007] consider two feature sets: terms and 2-word proximity expressions. In theory, $\mathcal{E}$ should include all expressions. In practice, only the expressions with the highest $P_{H,\Lambda}(e|Q)$ are considered for $\mathcal{Z}_Q$.

These expression weights are used to construct a second, weighted query. The document scores after a second retrieval are computed using a combined query,

$$P_{H,\Lambda}(D|Q') = \exp(\underbrace{F_{DQ'}(D,Q')}_{\text{expansion}})^{\alpha} \times \exp(\underbrace{F_{DQ}(D,Q)}_{\text{original query}})^{(1-\alpha)}$$

Claim 6.5. *Latent concept expansion is a form of regularization.*

Proof. Beginning with the ranking function for the expanded query,

$$\begin{aligned}
P_{H,\Lambda}(D|Q') &= \exp(F_{DQ'}(D,Q'))^{\alpha} \times \exp(F_{DQ}(D,Q))^{(1-\alpha)} \\
&= \exp\left(\sum_{e \in \mathcal{E}} \frac{P_{H,\Lambda}(e|Q)}{\mathcal{Z}_Q} \log P(e|D)\right)^{\alpha} \times \exp\left(\log P_{G,\Lambda}(D|Q)\right)^{(1-\alpha)} \\
&\overset{\text{rank}}{=} \alpha \sum_{e \in \mathcal{E}} \frac{P_{H,\Lambda}(e|Q)}{\mathcal{Z}_Q} \log P(e|D) + (1-\alpha) \log P_{G,\Lambda}(D|Q) \\
&= \alpha \sum_{e \in \mathcal{E}} \frac{1}{\mathcal{Z}_Q} \left[\sum_{D' \in \mathcal{R}_Q} P_{G,\Lambda}(D'|Q) \frac{\tilde{P}(e|D')^{\lambda'_{T_D}}}{\tilde{P}(e|C)^{\lambda'_{T_Q}}} \right] \log P(e|D) + (1-\alpha) \log P_{G,\Lambda}(D|Q) \\
&= \alpha \sum_{D' \in \mathcal{R}_Q} \frac{P_{G,\Lambda}(D'|Q)}{\mathcal{Z}_Q} \left[\sum_{e \in \mathcal{E}} \frac{\tilde{P}(e|D')^{\lambda'_{T_D}}}{\tilde{P}(e|C)^{\lambda'_{T_Q}}} \log P(e|D) \right] + (1-\alpha) \log P_{G,\Lambda}(D|Q)
\end{aligned}$$

If we let y be the length n vector of original document scores such that $y_i = P_{G,\Lambda}(d_i|Q)$, then we can define the updated scores, f, such that

$$f = \frac{\alpha}{\|y\|_1} Ay + (1-\alpha)\log(y)$$

where A is like an idf-weighted cross entropy defined as

$$A_{ij} = \sum_{e \in \mathcal{E}} \frac{\tilde{P}(e|D_j)^{\lambda'_{T_D}}}{\tilde{P}(e|C)^{\lambda'_{T_Q}}} \log P(e|D_i)$$

This means that LCE is theoretically equivalent to a single step of iterative regularization using a concept-based similarity measure. □

Metzler and Croft indicate that expanding using multi-term expression never improved retrieval performance above expansion by single terms. This reduction suggests one possible explanation: the accuracy of inter-document similarity measures is usually not improved by considering more complicated features. This is consistent with the insignificant gains bigrams see in classification and link detection tasks [Bekkerman and Allan, 2004; Nallapati and Allan, 2002].

6.4 Laplacian Eigenmaps

Score regularization can be viewed as nonparametric function approximation. An alternative method of approximation reconstructs $\mathbf{y}$ with smooth basis functions. When put in this perspective, reconstructing the original function, $\mathbf{y}$, using smooth basis functions indirectly introduces the desired inter-document consistency [Belkin and Niyogi, 2003]. When Fourier analysis is generalized to the discrete situation of graphs, the eigenvectors of Δ provide a set of orthonormal basis functions. We can then construct a smooth approximation of $\mathbf{y}$ using these basis functions. In this situation, our solution is,

$$\mathbf{f}^* = \mathbf{E}\left(\mathbf{E}^\mathsf{T}\mathbf{E}\right)^{-1}\mathbf{E}^\mathsf{T}\mathbf{y} \tag{6.11}$$

where $\mathbf{E}$ is a matrix consisting of the k eigenvectors of Δ associated with the smallest k eigenvalues. These eigenvectors represent the low frequency harmonics on the graph and therefore result in smooth reconstruction.[3]

Claim 6.6. *Function approximation using harmonic functions of the document graph is a form of regularization.*

Proof. We can view this process from the perspective of cluster-based retrieval. In the vector space model, Equation 6.11 can be rewritten as,

$$\begin{aligned}\mathbf{f}^* &= \mathbf{E}\left(\mathbf{E}^\mathsf{T}\mathbf{E}\right)^{-1}\mathbf{E}\mathbf{C}\mathbf{q} \\ &= \left[\mathbf{E}\left(\mathbf{E}^\mathsf{T}\mathbf{E}\right)^{-1}\right]\left[\mathbf{E}^\mathsf{T}\mathbf{C}\right]\mathbf{q} \\ &= \mathbf{V}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{q}\end{aligned} \tag{6.12}$$

[3] We note that although harmonic reconstruction has been successfully used for text classification tasks [Belkin and Niyogi, 2003], in our experience, the approach does not produce significant improvements for retrieval. This result follows from the fact that the retrieval score functions are likely to be much more peaked than classification score functions (cf. Figure 2.1). Szlam *et al.* note [Szlam et al., 2006, p. 3], "The Fourier modes ϕ_1 are global functions, and hence the projection of a function f onto the top eigenvectors of the diffusion operator is affected by global properties of the space and f , and may destroy important local features of f."

where the $k \times m$ matrix $\mathbf{U}^\mathsf{T}$ represents the basis as *linear* combinations of document vectors and the $n \times k$ matrix $\mathbf{V}^\mathsf{T}$ projects documents into the lower dimensional space. In language model retrieval, Equation 6.11 can be rewritten as,

$$\begin{aligned} \mathbf{f}^* &= \mathbf{E}\left(\mathbf{E}^\mathsf{T}\mathbf{E}\right)^{-1}\mathbf{E}\log(\mathbf{C})\mathbf{q} \\ &= \left[\mathbf{E}\left(\mathbf{E}^\mathsf{T}\mathbf{E}\right)^{-1}\right]\left[\mathbf{E}\log(\mathbf{C})\right]\mathbf{q} \\ &= \mathbf{V}^\mathsf{T}\log(\mathbf{U}^\mathsf{T})\mathbf{q} \end{aligned} \tag{6.13}$$

where the $k \times m$ matrix $\mathbf{U}^\mathsf{T}$ represents the eigenfunctions as *geometric* combinations of document vectors.

In both situations, new scores are computed as functions of cluster scores and cluster affinities. Therefore, we claim that basis reconstruction methods are an instance of score regularization. □

6.5 Link Analysis Algorithms

Graph representations often suggest the use discrete metrics such as PageRank to reweight initial retrieval scores [Brin and Page, 1998; Cohn and Hofmann, 2000; Kleinberg, 1998; Kurland and Lee, 2005]. These metrics can be thought of as functions from a document to a real value, $g_\mathbf{W} : \mathcal{D} \rightarrow \Re$. The function is indexed by the weight matrix $\mathbf{W}$ because these metrics are often dependent only on the graph structure. Let $\mathbf{g}$ be the length-$\tilde{n}$ vector of values of g for our $\tilde{n}$ documents. We will refer to this vector as the *graph structure function*. The values in $\mathbf{g}$ are often combined with those in $\mathbf{y}$ by linear combination (eg, $\mathbf{f} = \mathbf{y} + \mathbf{g}$) or geometric combination (eg, $\mathbf{f} = \mathbf{y} \circ \mathbf{g}$).

Many of these methods are instances of the spectral techniques presented in Section 6.4 [Ng et al., 2001]; specifically, PageRank is the special case where only the top eigenvector is considered (ie, $\mathbf{g} = \mathbf{E}_1$).

We believe it is very important to ask why the graph represented in $\mathbf{W}$ is being used in retrieval. For regularization, the matrix $\mathbf{W}$ by design enforces inter-document score consistency. For hypertext, the matrix $\mathbf{W}$ (by way of $\mathbf{g}$) provides the stationary distribution of the Markov chain defined by the hypertext graph. This can be a good model of page popularity in the absence of true user visitation data. When better user visitation information is available, though, the model provided by $\mathbf{g}$ is less useful [Richardson et al., 2006]. When the graph $\mathbf{W}$ is derived from content-based similarities, what does $\mathbf{g}$ mean? It is unclear that content-derived links can be navigational surrogates; the hypothesis has never been tested. Therefore, applications of graph structure functions to content-based graphs seem weakly justified. We believe that the incorporation of graph structure through regularization, by contrast, has a more solid theoretical motivation.

Because the structure information is lost when computing $\mathbf{g}$ from $\mathbf{W}$, we cannot claim that link analysis algorithms are an instance of regularization.

6.6 Spreading Activation

When viewed as a diffusion algorithm, our work is also related to the many spreading activation algorithms [Belew, 1989; Kwok, 1989; Salton and Buckley, 1988; Wilkinson and Hingston, 1991; Croft et al., 1988] and inference network techniques [Turtle and Croft, 1990; Metzler and Croft, 2004]. In these systems, terms and documents form a bipartite graph. Usually only direct relationships such as authors or sources allow inter-document links. These algorithms operate on functions from nodes to real values, $h : \{\mathcal{D} \cup \mathcal{V}\} \rightarrow \Re$. The domain of the functions includes both documents and terms. The domain of the functions in regularization includes only documents. Clearly spreading activation is not a form of regularization.

However, since regularization is a subset of spreading activation techniques, why should we study it on its own? First, it is not clear that the smoothness objective is appropriate for heterogeneous graphs. The assertion that the scores of a term and a document should be comparable is tenuous. Second, we believe that our perspective is theoretically attractive because of its ability to bring together several pseudo-relevance feedback techniques under a single framework. Nevertheless, the formal study of heterogeneous nodes in a manner similar to score regularization is a very interesting area of future work.

6.7 Relevance Propagation

Hypertext collections have inspired several algorithms for spreading content-based scores over the web graph [Qin et al., 2005]. These algorithms are equivalent to using a hyperlink-based affinity matrix and iterative regularization. A similar approach for content-based affinity has also been proposed [Savoy, 1997]. The foundation of these algorithms is at times heuristic, though. We believe that our approach places regularization—whether based on hyperlinks or content affinity—in the context of a mathematical formalism.

6.8 Summary

In this chapter, we have studied methods which directly and indirectly exploit corpus structure. In particular, we have examined these methods from the perspective of score regularization. We present a summary of these results in Table 6.1.

In the course of our derivations, we have sought to generalize and squint when necessary to show similarities between algorithms. In practice, the implementation of these algorithms differs from what is presented here. We believe these implementation differences explain some performance differences and deserve more detailed analysis.

A variety of graph algorithms exist which use links based on content and hyperlinks. These algorithms often are very subtle variations of each other when analyzed. We hope that our discussion will provide a basis for comparing graph-based and corpus structure algorithms for information retrieval.

We have restricted our discussion of scoring algorithms to two popular approaches: vector space retrieval and retrieval of language models. Certainly other models exist and deserve similar treatment. This chapter should provide a perspective not on only analyzing

	score
Vector Space Model	
Query Expansion	$\mathbf{Ay} + \mathbf{y}$
Document Expansion	$\mathbf{Wy} + \mathbf{y}$
Cluster-based Retrieval	$\mathbf{V}^\mathsf{T}\mathbf{y}_c + \mathbf{y}$
Language Modeling	
Query Expansion	$\mathbf{Ay} + \mathbf{y}$
Document Expansion	$\log(\mathbf{AC} + \mathbf{C})\mathbf{q}$
Cluster-based Retrieval	$\log(\mathbf{V}^\mathsf{T}\mathbf{U}^\mathsf{T} + \mathbf{C})\mathbf{q}$
Cluster Interpolation	$\mathbf{W}_e\mathbf{y}_e + \mathbf{y}$
Feature-Based Retrieval	
Query Expansion	$\mathbf{Ay} + \mathbf{y}$
Regularization	
Iterative Regularization	$\mathbf{Wy} + \mathbf{y}$
Closed Form Regularization	$(\alpha\Delta + (1-\alpha)\mathbf{I})^{-1}\mathbf{y}$
Laplacian Eigenmaps	$\mathbf{W}_c\mathbf{y}_c$
PageRank	$\mathbf{E}_1 \circ \mathbf{y}$

Table 6.1. Comparison of corpus modeling and graph-based algorithms. Model-specific constants and parameters have been omitted for clarity.

query expansion, regularization, and document expansion in other frameworks but also on developing query expansion, regularization, and document expansion for new frameworks.

Finally, we believe that the results of this chapter indicate that the improvement of local score consistency explains some of the success of previous approaches. However, we note that few of these approaches *directly* incorporate consistency, often opting instead for application of what amounts to a single iteration of regularization. We further believe that the direct and formal incorporation of consistency provides a compelling area for future work.

CHAPTER 7

STABILITY OF REGULARIZATION

The fundamental data structure in our regularization algorithm is the inter-document affinity matrix. According to van Rijsbergen, text affinity measures share heuristics which result in very similar behavior [van Rijsbergen, 1979, page 24],

> Informally speaking, a measure of association increases as the number or proportion of shared attribute states increases. Numerous coefficients of association have been described in the literature, see for example Goodman and Kruskal, Kuhns, Cormack and Sneath and Sokal. Several authors have pointed out that the difference in retrieval performance achieved by different measures of association is insignificant, providing that these are appropriately normalised. Intuitively, one would expect this since most measures incorporate the same information. Lerman has investigated the mathematical relationship between many of the measures and has shown that many are monotone with respect to each other. It follows that a cluster method depending only on the rank-ordering of the association values would given identical clusterings for all these measures.

In Section 2.3, we described severals ways to define this matrix. In this chapter, we will establish theoretical bounds and present empirical evidence of the effect different similarity measures have on the stability of regularization.

We will view regularization as the solution of a linear system. We rewrite the closed form version of regularization (Equation 5.11) as,

$$\left(\frac{\alpha}{1-\alpha}\Delta + \mathbf{I}\right)\mathbf{f}^* = \mathbf{y} \tag{7.1}$$

where the Laplacian, Δ, is associated with the matrix, $\mathbf{W}$, generated by some arbitrary similarity measure (for example, cosine similarity).

We consider a matrix, $\tilde{\mathbf{W}}$, generated by a different similarity matrix (for example Hellinger similarity). The regularized scores using this alternative similarity measure is the solution to the linear system in Equation 7.1 using $\tilde{\mathbf{W}}$. Let $\tilde{\Delta}$ be the Laplacian of $\tilde{\mathbf{W}}$. The linear system for the perturbed matrix is,

$$\left(\frac{\alpha}{1-\alpha}\tilde{\Delta} + \mathbf{I}\right)\tilde{\mathbf{f}}^* = \mathbf{y} \tag{7.2}$$

We would like to bound the difference in regularized scores given differences in the similarity matrix. We will measure the change in regularized scores using the relative error between scores,

$$\frac{\|\tilde{\mathbf{f}}^* - \mathbf{f}^*\|_2}{\|\mathbf{f}^*\|_2} \tag{7.3}$$

We will measure the difference in the similarity matrix according to the changes in the associated Laplacians,

$$\left\|\tilde{\Delta} - \Delta\right\|_2 \tag{7.4}$$

where the matrix norm is induced from the vector 2-norm.

Theorem 7.1.

$$\frac{\|\tilde{\mathbf{f}}^* - \mathbf{f}^*\|}{\|\mathbf{f}^*\|} \leq \frac{\alpha}{1-\alpha}\left\|\tilde{\Delta} - \Delta\right\|$$

Proof. We will be treating the solution in Equation 7.2 as the solution to a perturbed version of Equation 7.1 [Stewart and Sun, 1990]. To this end, we rewrite Equation 7.2 to show the perturbation more explicitly,

$$(\mathbf{A} + \mathcal{E})\,\tilde{\mathbf{f}}^* = \mathbf{y}$$

where

$$\begin{aligned}
\mathbf{A} &= \frac{\alpha}{1-\alpha}\Delta + \mathbf{I} \\
\mathcal{E} &= \frac{\alpha}{1-\alpha}\left(\tilde{\Delta} - \Delta\right) \\
\tilde{\mathbf{A}} &= \mathbf{A} + \mathcal{E} \\
&= \frac{\alpha}{1-\alpha}\tilde{\Delta} + \mathbf{I}
\end{aligned}$$

The difference between solutions is then defined as,

$$\begin{aligned}
\mathbf{f}^* - \tilde{\mathbf{f}}^* &= \mathbf{A}^{-1}\mathbf{y} - \tilde{\mathbf{A}}^{-1}\mathbf{y} \\
&= (\mathbf{A}^{-1} - \tilde{\mathbf{A}}^{-1})\mathbf{y}
\end{aligned}$$

Because $\mathbf{A}$ and $\tilde{\mathbf{A}}$ are nonsingular,

$$\begin{aligned}
\tilde{\mathbf{A}}^{-1}\tilde{\mathbf{A}} &= \mathbf{I} \\
\tilde{\mathbf{A}}^{-1}\mathbf{A} + \tilde{\mathbf{A}}^{-1}\mathcal{E} &= \mathbf{A}^{-1}\mathbf{A} \\
\tilde{\mathbf{A}}^{-1} + \tilde{\mathbf{A}}^{-1}\mathcal{E}\mathbf{A}^{-1} &= \mathbf{A}^{-1} \\
\tilde{\mathbf{A}}^{-1}\mathcal{E}\mathbf{A}^{-1} &= \mathbf{A}^{-1} - \tilde{\mathbf{A}}^{-1}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbf{f}^* - \tilde{\mathbf{f}}^* &= \tilde{\mathbf{A}}^{-1}\mathcal{E}\mathbf{A}^{-1}\mathbf{y} \\
&= \tilde{\mathbf{A}}^{-1}\mathcal{E}\mathbf{f}^* \\
\|\mathbf{f}^* - \tilde{\mathbf{f}}^*\| &\leq \|\tilde{\mathbf{A}}^{-1}\|\|\mathcal{E}\|\|\mathbf{f}^*\| \\
\frac{\|\mathbf{f}^* - \tilde{\mathbf{f}}^*\|}{\|\mathbf{f}^*\|} &\leq \|\tilde{\mathbf{A}}^{-1}\|\|\mathcal{E}\|
\end{aligned} \tag{7.5}$$

Now, we compute the value of $\|\tilde{\mathbf{A}}^{-1}\|$. First, we can show that $\tilde{\mathbf{A}}$ is positive definite. That is, $\mathbf{x}^\mathsf{T}\tilde{\mathbf{A}}\mathbf{x} > 0$ for all $\mathbf{x} > 0$. The proof is rather straightforward, using the fact that the Laplacian is positive semidefinite [Chung, 1997],

$$\begin{aligned}
\mathbf{x}^\mathsf{T}\tilde{\mathbf{A}}\mathbf{x} &= \mathbf{x}^\mathsf{T}\left(\frac{\alpha}{1-\alpha}\tilde{\Delta} + \mathbf{I}\right)\mathbf{x} \\
&= \frac{\alpha}{1-\alpha}\mathbf{x}^\mathsf{T}\tilde{\Delta}\mathbf{x} + \mathbf{x}^\mathsf{T}\mathbf{x} \\
&\geq \mathbf{x}^\mathsf{T}\mathbf{x} \\
&> 0
\end{aligned}$$

Given this, we know that, for positive definite matrices, $\|\tilde{\mathbf{A}}^{-1}\| = \frac{1}{\lambda_{min}(\tilde{\mathbf{A}})}$, where $\lambda_{min}(\tilde{\mathbf{A}})$ is the minimum eigenvalue of $\tilde{\mathbf{A}}$. We can derive the following relationship between the eigenvalues of $\tilde{\mathbf{A}}$ and $\tilde{\Delta}$,

$$\begin{aligned}
\tilde{\mathbf{A}}\mathbf{x} &= \lambda\mathbf{x} \\
\left(\frac{\alpha}{1-\alpha}\tilde{\Delta} + \mathbf{I}\right)\mathbf{x} &= \lambda\mathbf{x} \\
\frac{\alpha}{1-\alpha}\tilde{\Delta}\mathbf{x} + \mathbf{x} &= \lambda\mathbf{x} \\
\frac{\alpha}{1-\alpha}\tilde{\Delta}\mathbf{x} &= (\lambda - 1)\mathbf{x} \\
\tilde{\Delta}\mathbf{x} &= \frac{(\lambda-1)(1-\alpha)}{\alpha}\mathbf{x}
\end{aligned}$$

The minimum eigenvalue of the Laplacian is 0 [Chung, 1997]. Therefore, the minimum eigenvalue of $\tilde{\mathbf{A}}$ is 1 and $\|\tilde{\mathbf{A}}^{-1}\| = 1$. Substituting this value into Equation 7.5 completes our proof.

□

Remark 7.1. $\|\tilde{\Delta} - \Delta\| \leq 2$

Proof. Because Δ and $\tilde{\Delta}$ are symmetric, by Fischer's theorem [Stewart and Sun, 1990, Corollary IV.4.7], we can establish the following bound on the norm of their difference,

$$\begin{aligned}
\left\|\tilde{\Delta} - \Delta\right\| &= \lambda_{max}(\tilde{\Delta} - \Delta) \\
&\leq \lambda_{max}(\tilde{\Delta}) + \lambda_{max}(-\Delta) \\
&= \lambda_{max}(\tilde{\Delta}) + 0 \\
&\leq 2
\end{aligned}$$

□

We depict the bound on $\frac{\|\tilde{\mathbf{f}}^* - \mathbf{f}^*\|_2}{\|\mathbf{f}^*\|_2}$ in Figure 7.1. The general behavior of this bound confirms an intuition we might have already. For low values of α, when two affinity measures are very similar, their regularized scores are very similar. In fact, for low values of α, the

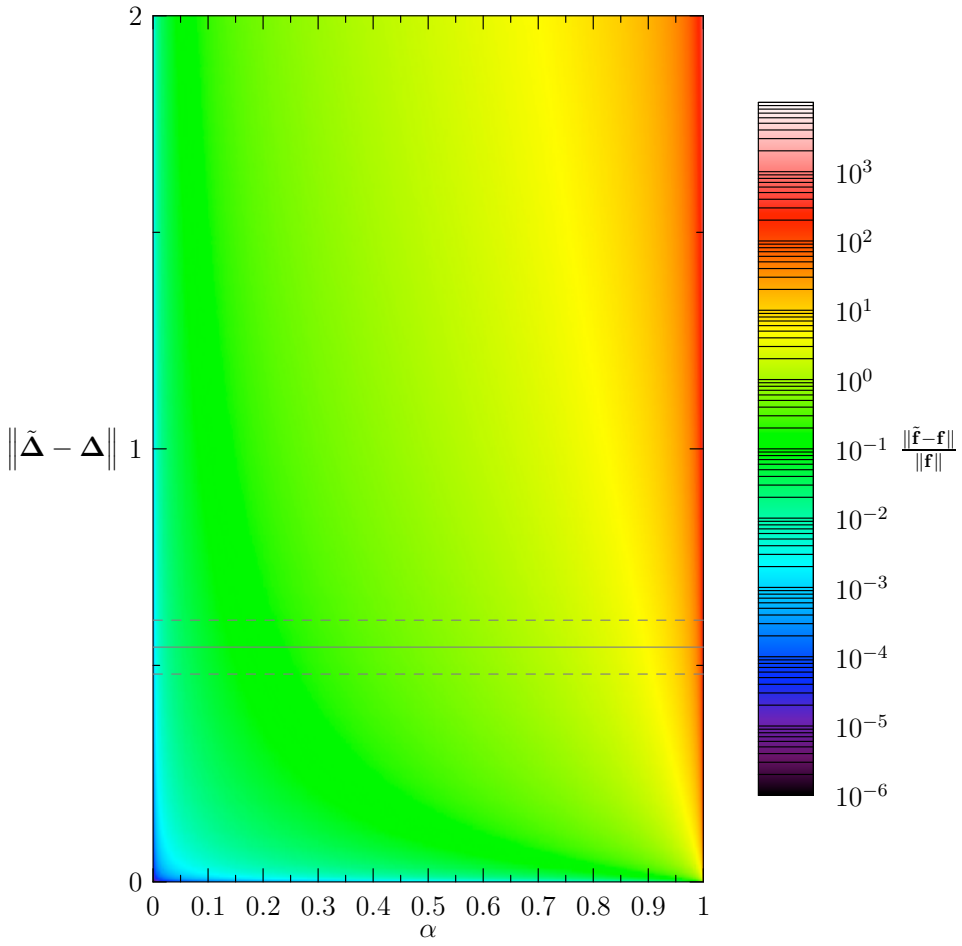


Figure 7.1. Bound on regularization error given similarity matrix perturbations and α. The solid horizontal line represents the empirical mean perturbation found in our experiments. The dashed lines represent one standard deviation. This graph is ideally viewed in color.

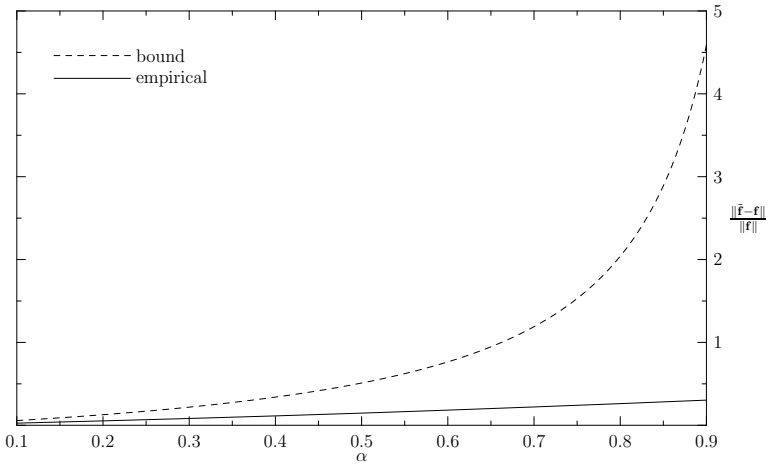


Figure 7.2. Empirical differences in regularized scores as a function of α for a retrieval from our experiments. This dashed line in this graph represents the theoretical bound and therefore is a cross-section of surface from Figure 7.1.

regularization is quite robust to arbitrary differences in the affinity. However, as we regularize more aggressively by using a higher α, the regularized solutions are more sensitive to perturbations of the affinity matrix.

The range of differences, $0 \leq \|\tilde{\Delta} - \Delta\| \leq 2$, includes arbitrary matrix perturbations. In reality, our perturbations are likely to be constrained to differences much less than the maximum.[1] In order to measure the empirical perturbations, we computed the differences between Laplacians using cosine similarity and the Hellinger distance over all retrievals in trec12/QL. We found that the mean value of $\|\tilde{\Delta} - \Delta\|$ was 0.541 ± 0.0585; this range is depicted in Figure 7.1. An expected perturbation in this range indicates that regularized scores will be very similar for $0 \leq \alpha \leq 0.5$. We plot the empirical regularization differences for various α for a fixed query with $\|\tilde{\Delta}-\Delta\| = 0.510$ in Figure 7.2. From this figure, it should be clear that our bound, because it considers arbitrary perturbations, is somewhat loose. The empirical evidence from other queries indicates that the actual differences between these two affinity measures is likely to be far below our bound.

The bound established in Theorem 7.1 measures the effect of perturbations on norm of the difference between the regularized scores. Because information retrieval is often evaluated by the induced ranking, it is worth exploring the effect on rankings resulting from perturbations. Therefore, for each pair of regularized rankings in our experiment, we compute the Plantagenet coefficient of rank similarity [Genest and Plante, 2003]. The Plantagenet is defined as,

[1] Because we are constraining our analysis to *symmetric* matrices perturbed by *symmetric* matrices, we might believe that the bound is in fact much smaller. However, Higham proved that such constraints in fact do not change the condition number and therefore we suspect that symmetric perturbation *may* not affect our bound [Higham, 1995].

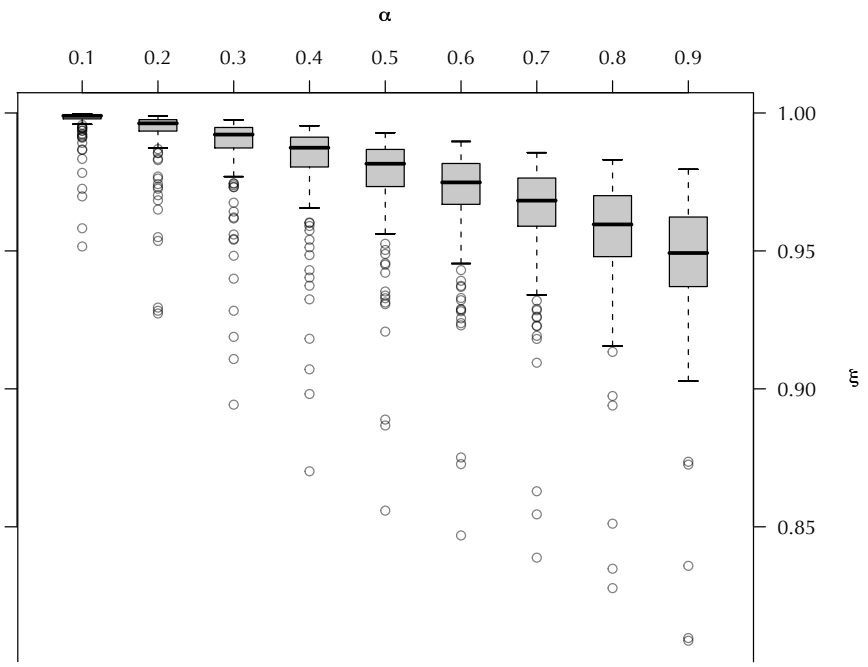


Figure 7.3. Empirical relationship between α and the Plantagenet coefficient.

$$\xi_n = -\frac{4n+5}{n-1} + \frac{6}{n^3-n}\sum_{i=1}^{n} x_i y_i \left(4 - \frac{x_i + y_i}{n+1}\right) \tag{7.6}$$

where **x** and **y** are vectors containing ranks of each document. The Plantagenet coefficient is a version of Spearman correlation sensitive to changes in the top ranks. This measure is appropriate because, when comparing two rankings, we are most concerned with changes of the ranks of top-ranked documents. In Figure 7.3, we plot this correlation as a function of α. This figure gives us an intuition for the perceptible changes resulting from using different similarity measures. In fact, we see that, for all values of α, we achieve strong correlation between rankings. This indicates that $\frac{\|\tilde{\mathbf{f}}^* - \mathbf{f}^*\|}{\|\mathbf{f}^*\|}$ gives us an accurate representation of the effect of affinity perturbation.

Finally, we can also measure the effect of perturbation on differences in performance. In Figure 7.4, we plot the relative differences in average precision resulting from changes in the affinity measure. We notice that the mean relative change in average precision is less than 10% for all values of α. This again indicates that regularization is, on average, not sensitive to the similarity measure. Nevertheless, some of the outlying queries seem to be more sensitive to α than the mean.

In summary, we have studied the stability of regularization subject to changes of the parameter α. We found that, for small values of α, solutions are robust to small perturbations in the similarity matrix. For more aggressive regularization, solution are more sensitive to perturbations in the similarity matrix. We complemented these theoretical results with em-

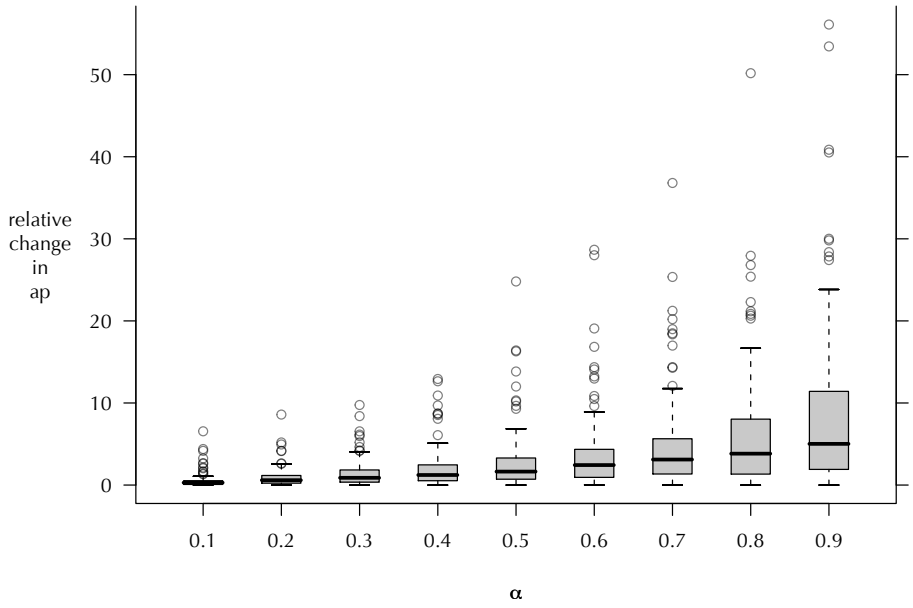


Figure 7.4. Empirical relationship between α and the relative change in average precision. The performance using cosine similarity are used as the baseline.

pirical measurements of the effect of similarity matrix perturbations. We found that the differences between vector space model and language model similarities resulted only in slight differences in regularized scores, rank ordering of regularized scores, and performance.

CHAPTER 8

EXTENSIONS AND FUTURE WORK

We have presented a theoretical and experimental analysis of score regularization. In this chapter, we will describe several extensions to the framework which demonstrate its applicability to relevance feedback, cross-lingual retrieval, optimal set retrieval, and cross-media retrieval.

8.1 Relevance Feedback

So far in this thesis, a system has been evaluated based on a single retrieval given a short query. In some situations, the user also supplies sample relevant and non-relevant documents. These judgments may be provided with the query or in response to some initial query probing the collection for documents. The second scenario, referred to as *relevance feedback*, will be the focus of this section.

All of the retrieval models in Section 2.2 have different methods for incorporating relevance judgments in interactive retrieval. We will be focusing on the language modeling approach. In Equation 2.12, when relevance judgments are absent, we estimate a language model of relevance using a weighted combination of documents in an initial retrieval. When document relevance information is provided, we can estimate the *true relevance model* directly with binary weights [Lavrenko, 2004, p. 69],

$$P(w|\theta_R) = \lambda P(w|\theta_Q) + (1-\lambda) \sum_{d \in \mathcal{R}^+} \frac{1}{|\mathcal{R}^+|} P(w|\theta_d)$$

where $\mathcal{R}^+$ is the set of documents judged relevant. In matrix notation, we represent this as

$$\tilde{\mathbf{q}} = \lambda \mathbf{q} + \frac{(1-\lambda)}{|\mathcal{R}^+|} \mathbf{C}^\mathsf{T} \mathbf{r} \tag{8.1}$$

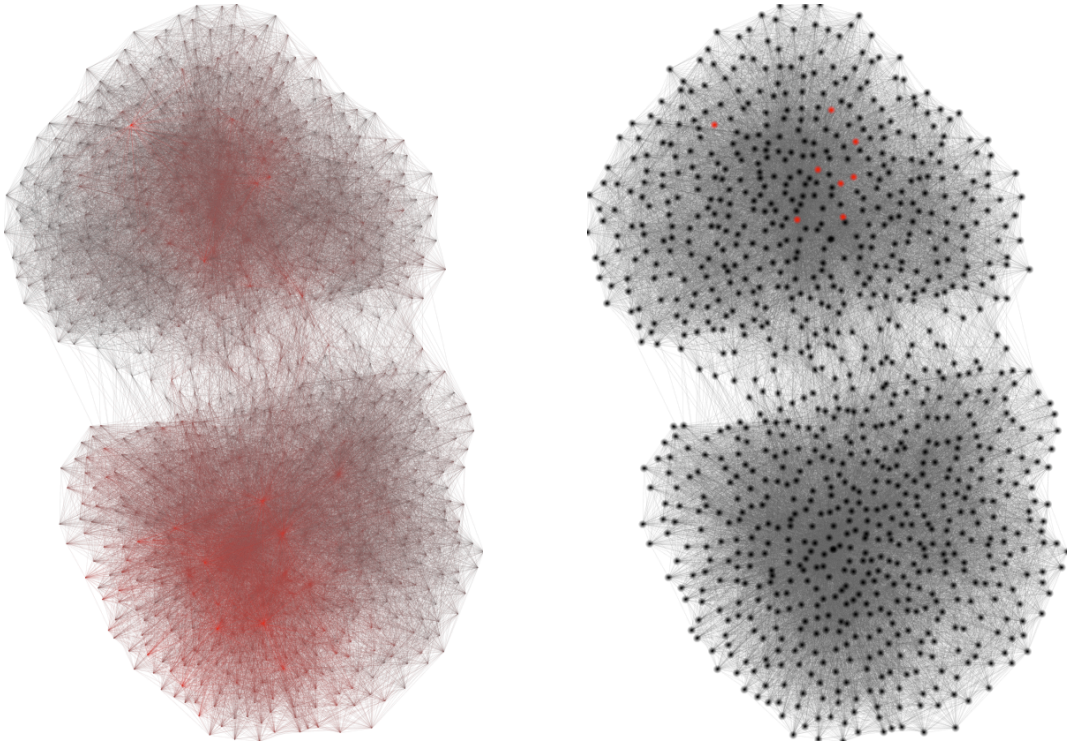
where $\mathbf{r}$ is an $n \times 1$ vector where,

$$r_i = \begin{cases} 1 & \text{if } i \in \mathcal{R}^+ \\ 0 & \text{otherwise} \end{cases} \tag{8.2}$$

Documents are then ranked according to cross entropy,

$$\mathbf{f} = \log(\mathbf{C}) \tilde{\mathbf{q}} \tag{8.3}$$

We note that there is no formal model of *non-relevance* in relevance feedback based on true relevance models. True relevance models approach information retrieval from the perspective of density estimation. Relevant examples provide statistics for the true relevance



(a) initial score function (b) relevant documents

Figure 8.1. A high-scoring non-relevant cluster. The figure on the left depicts the scores on the document graph. On the right, we show the relevance of each document. Red nodes indicate relevant documents. Black nodes indicate non-relevant documents.

model. The non-relevance model, by default, is the language model estimated using collection statistics, $P(w|\theta_C)$. As a naïve approach to incorporating non-relevant documents, we might add them to the documents used to model non-relevance. However, since the majority of the collection is non-relevant, the information from additional non-relevant documents would be washed out. This might be seen as a minor theoretical detail given the empirical evidence that negative feedback does not result in significant improvements [Aalbersberg, 1992; Dunlop, 1997]. However, we believe that the information in explicitly non-relevant documents can useful in situations where no relevant documents are retrieved and the system must filter non-relevant information. For example, if the only known keywords for a topic retrieve a cohesive, non-relevant cluster, we would like to provide information to remove the entire non-relevant cluster. We demonstrate this behavior in Figure 8.1. The higher-scoring but non-relevant documents fall into one cluster while lower-scoring but relevant documents fall into another cluster. Modeling non-relevance would allow the system to effectively down-weight all documents in the non-relevant cluster.

Whereas true relevance modeling is a non-parametric density estimation method, regularization is a non-parametric function approximation method. One advantage of approaching this as function approximation is that we can explicitly model non-relevance differently than we do uncertainty. Put another way, in Equation 8.2, a score of 0 represents

both non-relevant documents and unjudged documents. In regularization, if we normalize scores to zero mean and unit variance, we can explicitly model relevant document scores (eg, $y_i > 1$), non-relevant document scores (eg, $y_i < -1$), and unjudged documents (eg, $y_i = 0$).

In order to evaluate a relevance feedback method, we measure the performance of the system after receiving feedback on the top k documents. We evaluated the following model. After the user marks the top k documents as relevant or non-relevant, we re-issue the query using the true relevance model approach (Equation 8.1). We normalize the true relevance model scores, $\mathbf{y}$, to zero mean and unit variance. For each relevant document, we replace its score with a value sampled from the region of the Gaussian greater than the maximum score. We do the same replacement for each non-relevant document by using samples from the bottom region of the Gaussian. Given these adjusted scores, we perform our standard regularization.

We present the results of these experiments in Figure 8.2. Given the results in Chapter 5, we should not be surprised that regularization consistently improves the performance of retrieval. The interesting aspect of these plots is that the amount of improvement grows with the number of relevance judgments. We suspect that, as the number of judgments increases, the regularization component of the system becomes more important because the additional data introduces a more discriminative component to standard true relevance models, allowing us to take advantage of additional data [Ng and Jordan, 2002].

8.2 Cross-Lingual Regularization

Cross-lingual information retrieval refers to the task where a user is interested in documents written in a foreign or *target* language and provides a query in her native or *source* language. Traditional approaches to this problem usually perform some sort of query translation from the source to the target language. In this section, we will describe a technique for performing cross-lingual information retrieval without translating the query or performing a second retrieval. We refer to this technique as cross-lingual score regularization.

Formally, we have a target collection of n_t documents. Some small number, n_s, of the target documents have been translated into the source language. Sets of translated collections are common in the machine translation community and are referred to as *parallel corpora*. We will further assume that, given a query in the source language, we have some method for scoring the source language documents. For example, we might use one of the methods from Section 2.2.

8.2.1 Cross-Lingual Score Regularization

Cross-lingual regularization refers to the process of scoring the source parallel documents and then assuming that the n_s parallel target documents should have the same score. If the user were interested in retrieving the parallel target documents, the retrieval process could terminate at this stage. However, the user is more often interested in those target documents which do not have source translations. We will score these non-parallel target documents by

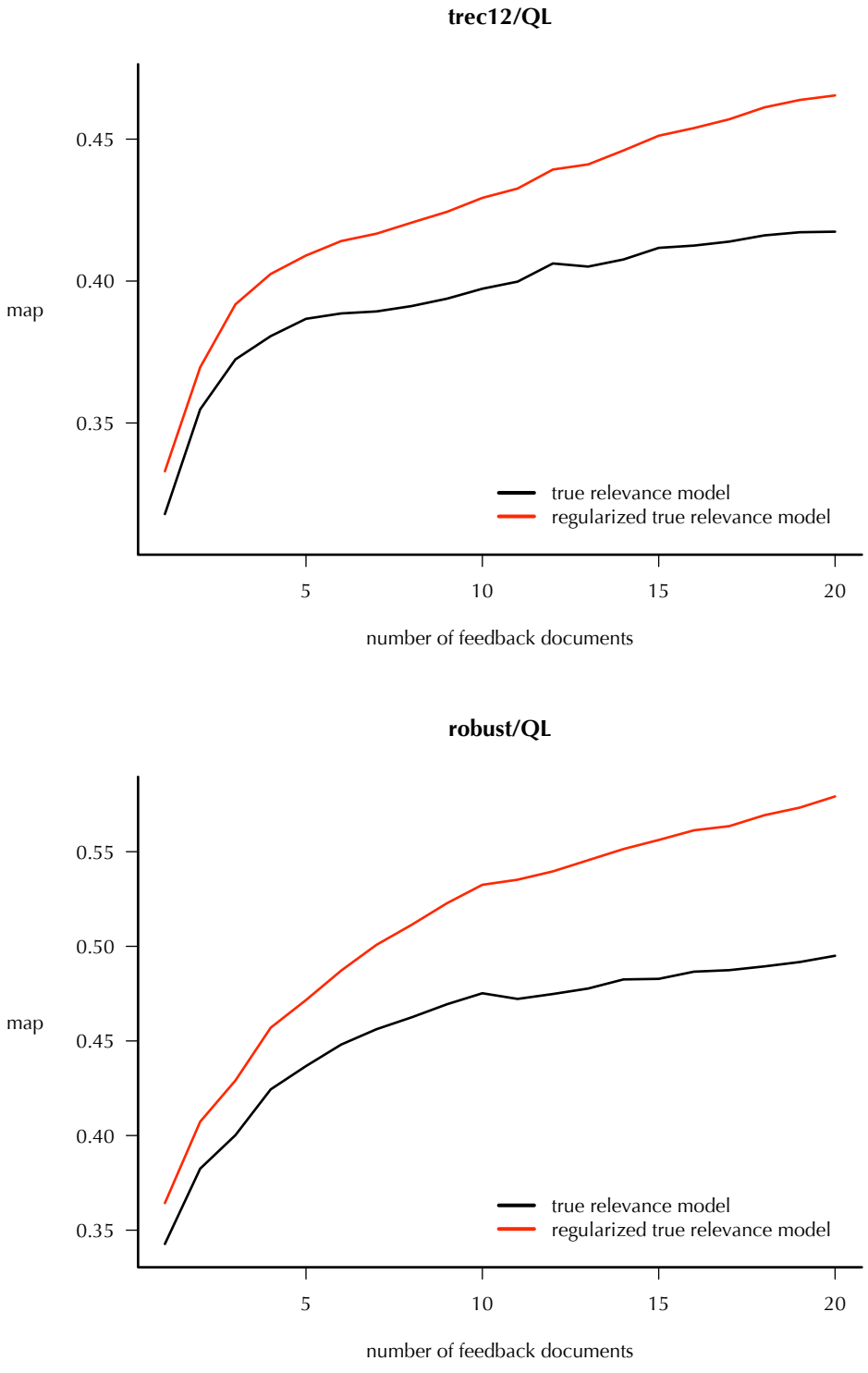


Figure 8.2. Relevance feedback results. The horizontal axis indicates the number of feedback documents judged from the initial retrieval. The vertical axis plots mean average precision of that retrieval. All regularized retrievals are rerankings of the true relevance model runs.

using the score information from the parallel documents. We depict this process graphically in Figure 8.3.[1]

Assume that the translated documents are all indexed identically from 0 to n_s for both corpora and that we have an $n_t \times n_t$ affinity matrix for the target collection of documents. Then the regularized target corpus scores are defined by the vector minimizing

$$\mathcal{Q}'(\mathbf{f}_t, \mathbf{y}_s) = \mathcal{S}'(\mathbf{f}_t) + \mu\mathcal{E}'(\mathbf{f}_t, \mathbf{y}_s) \tag{8.4}$$

where $\mathbf{y}_s$ is the $n_s \times 1$ vector of source collection scores and $\mathbf{f}_t$ is the $n_t \times 1$ vector of regularized target collection scores. The constraints are defined as,

$$\mathcal{S}'(\mathbf{f}_t) = \mathbf{f}_t^\mathsf{T} \Delta_t \mathbf{f}_t \tag{8.5}$$

$$\mathcal{E}'(\mathbf{f}_t, \mathbf{y}_s) = \mathbf{f}_t^\mathsf{T} \mathbf{y}_t \tag{8.6}$$

where $\mathbf{y}_t = [\mathbf{y}_s^\mathsf{T} \mathbf{0}^\mathsf{T}]^\mathsf{T}$ is a vector of projected scores. This problem has similar solutions to monolingual regularization. The iterative solution is,

$$\mathbf{f}_t^{t+1} = (1-\alpha)\mathbf{y}_t + \alpha \mathbf{S}_t \mathbf{f}_t^t \tag{8.7}$$

The closed form solution is,

$$\mathbf{f}_t^* = (1-\alpha)(\alpha\Delta_t + (1-\alpha)\mathbf{I})^{-1}\mathbf{y}_t \tag{8.8}$$

where $\alpha = \frac{1}{1+\mu}$.

8.2.2 Cross-Lingual Relevance Models

Let θ^a refer to a language model over the target vocabulary; similarly, θ^b models the source language. If we have a query in the target language, we score each target document, d, according to the query likelihood, $P(Q|\theta_d^b)$. The *cross-lingual relevance model* is estimated as,

$$P(w|\theta_R^a) = \sum_d \frac{P(Q|\theta_d^b)}{\mathcal{Z}} P(w|\theta_d^a) \tag{8.9}$$

The difference between the cross-lingual relevance model and the standard relevance model (Equation 2.12) is that we are applying the score for a source document to the parallel target document. This lets us build a relevance model in the target language using source document scores as the interpolation weights solving our problem of not having a query in the target language. In matrix notation,

$$\mathbf{q}_t = \frac{1}{\|\mathbf{y}_s\|_1} \mathbf{C}_t^\mathsf{T} \mathbf{y}_s$$

where $\mathbf{C}_t$ is our target collection and $\mathbf{y}_s$ is a $n_t \times 1$ vector where the n_s documents with translations are scored by $P(Q|\theta_d^b)$ and the $n_t - n_s$ target-only documents receive a score of 0. We can now use a cross-entropy scoring function to rank target documents,

$$\mathbf{f}_t = \log(\mathbf{C}_t)\mathbf{q}_t \tag{8.10}$$

[1] In the context of cross-lingual link detection, we used similar techniques successfully [Diaz and Metzler, 2007].

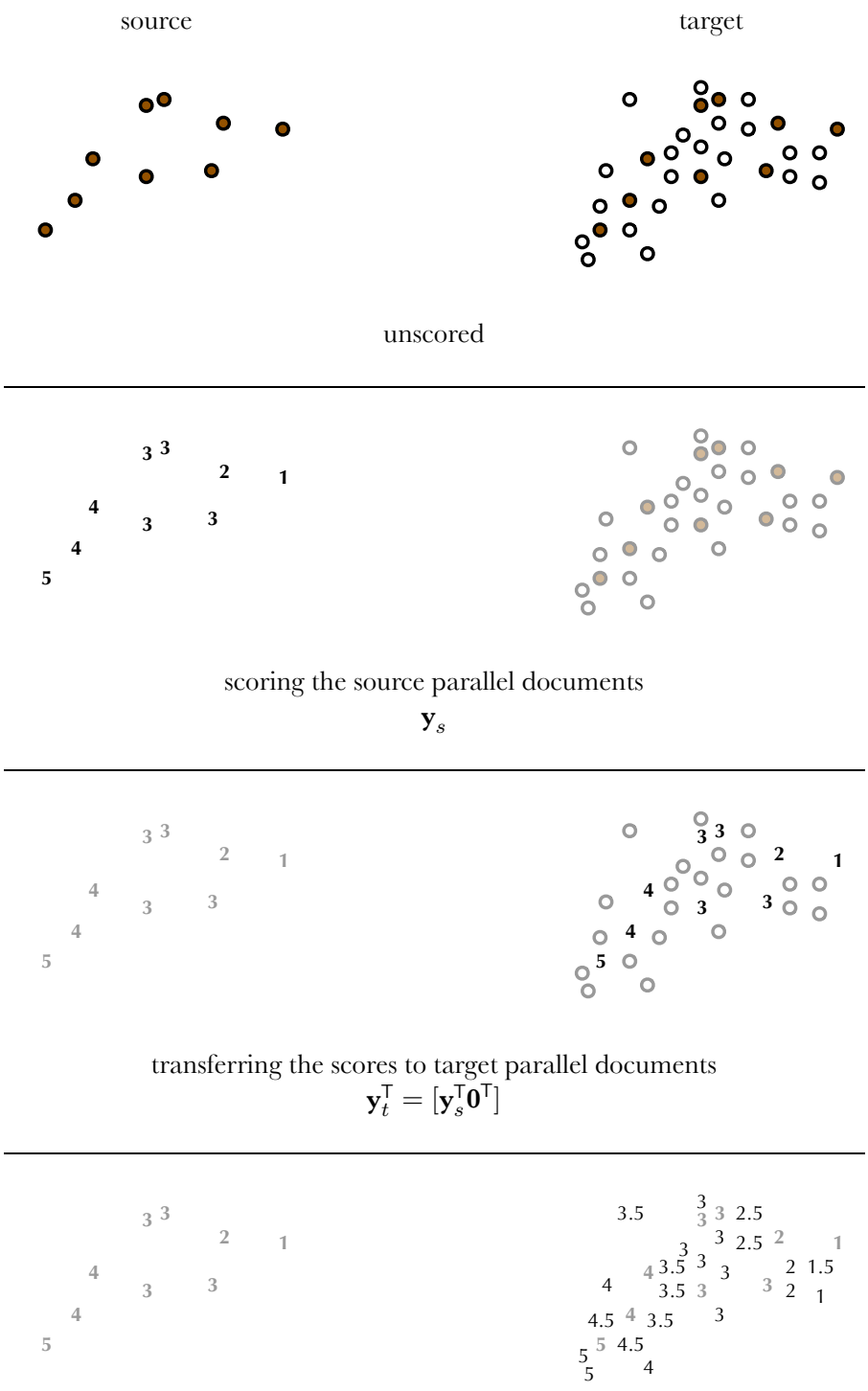


Figure 8.3. Cross-collection regularization by score projection. Documents in the parallel corpus are represented as brown circles. Documents of interest in the target language are represented as white circles. Bold numbers represent scores of the parallel documents. Unbolded numbers represent interpolated scores.

	CLRM	CLSR
0.00	0.5694	0.6238
0.10	0.3737	0.4456
0.20	0.3194	0.3535
0.30	0.2789	0.2943
0.40	0.2424	0.2502
0.50	0.2049	0.2010
0.60	0.1673	0.1520
0.70	0.1301	*0.0989*
0.80	0.0916	*0.0536*
0.90	0.0361	*0.0154*
1.00	0.0000	0.0000
map	0.2027	0.2064

Table 8.1. Cross-lingual relevance models compared to cross-lingual score projection.

Theorem 8.1. *Cross-lingual relevance models are a form of cross-lingual regularization.*

Proof. The proof follows that of Theorem 6.4. Starting from the ranking function,

$$\begin{aligned}\mathbf{y}_t &= \log(\mathbf{C}_t)\mathbf{q}_t \\ &= \frac{1}{\|\mathbf{y}_s\|_1}\log(\mathbf{C}_t)\mathbf{C}_t^{\mathsf{T}}\mathbf{y}_s \\ &\propto \mathbf{A}_t\mathbf{y}_s \end{aligned} \tag{8.11}$$

where $\mathbf{A}_t$ is an $n_t \times n_t$ affinity matrix based on inter-document cross-entropy between the target documents. □

8.2.3 Experiments

We compared the performance of cross-lingual score regularization (CLSR) to cross-lingual relevance models (CLRM) using a cross-lingual retrieval task involving a source query written in English and a target collection written in Mandarin [Smeaton, 1996].

We present results in Table 8.1. Perhaps due to the similarity of the approaches, there is no statistical difference between CLRM and CLSR. Upon investigation of the results, however, we notice that CLSR tends to perform better at the top of the ranked list while CLRM performs better at low-recall areas. To explore this further, we can look at the performance for high precision measures. In Table 8.2, we evaluate each algorithm by the precision for the top k documents. Although mostly statistically similar, CLSR performs significantly better when considering the top 10 documents.

P@K	CLRM	CLSR
5	0.3556	0.4185
10	0.3167	**0.4037**
15	0.3123	0.3617
20	0.3102	0.3389
30	0.3006	0.3228

Table 8.2. Cross-lingual relevance models compared to cross-lingual score projection.

8.3 Future Work

8.3.1 Optimal Cluster Retrieval

Sometimes the user is interested in being provided a high precision *set* of documents instead of a ranking of the entire collection [Dai and Srihari, 2005; Griffiths et al., 1986; Hearst and Pedersen, 1996; Jardine and van Rijsbergen, 1971; Liu and Croft, 2006]. This is important when the information retrieval user is an automatic process such as a text summarization system. Previous approaches to this task attempt to detect a single, tight cluster in an initial retrieval. An alternative approach, suggested by our score regularization framework, treats this as an optimization problem.

In regularization, we are concerned with document scores which induce a ranking; that is, $\mathbf{f} \in \Re^n$. In optimal cluster retrieval, we are concerned with document scores which induce a partition; that is $\mathbf{f} \in \{0, 1\}^n$ where documents with a score of 1 are retrieved. The principle objective of optimal cluster retrieval is that the retrieved set be on the same topic. One way of measuring this property is to inspect the local relationships in the set,

$$\frac{\mathbf{f}^\mathsf{T}\mathbf{W}\mathbf{f}}{\|\mathbf{f}\|} = \frac{\sum_{ij} W_{ij} f_i f_j}{\sum_i f_i^2}$$

When the value of this objective is small, documents in the set are unrelated to each other; when it is large, the documents in the set have high similarity. Notice that this is equal to the Moran autocorrelation of $\mathbf{f}$ (Equation 4.5). Although the similarity of documents within the set is important, we might alternatively be interested in the retrieved cluster being dissimilar from the rest of the corpus.

$$\frac{\mathbf{f}^\mathsf{T}\Delta_C\mathbf{f}}{\|\mathbf{f}\|} = \frac{\sum_{ij} W_{ij}(f_i - f_j)^2}{\sum_i f_i^2}$$

which is equivalent to a graph min-cut objective or—in spatial data analysis—the Geary autocorrelation [Cliff and Ord, 1973].

Unfortunately, these purely graph-based objectives ignore the relevance of the documents, $\mathbf{y}$, potentially resulting in retrieval of clusters of documents which are non-relevant. In Figure 8.4, we present a situation where ignoring the documents scores may lead to the selection of low-scoring documents. This figure also demonstrates that the optimal set may consist of documents from a portion of a cluster. In order to address this we can develop

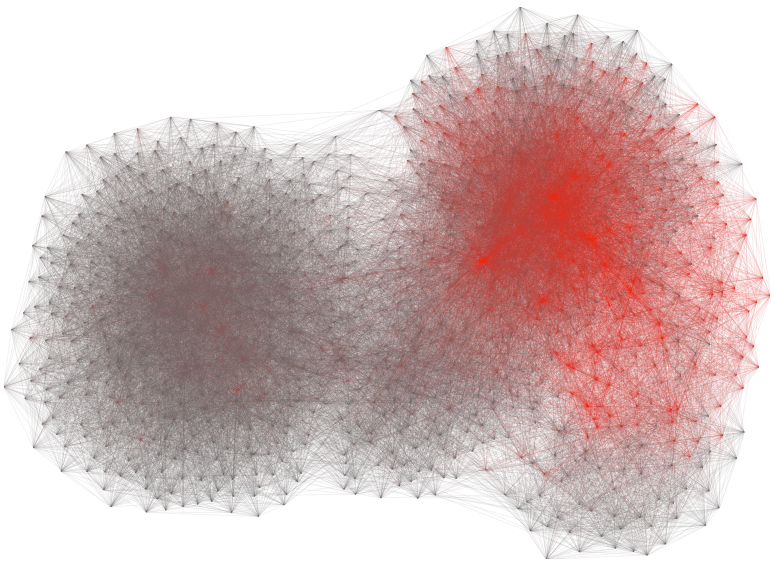


Figure 8.4. Retrieval function for the query "nuclear proliferation". This is a function which is not consistent with the topology of the document graph.

additional constraints on $\mathbf{f}$ that ensure that we select high-scoring documents. The easiest constraint would simply select a subset with a high average score,

$$\frac{\mathbf{f}^T\mathbf{y}}{\|\mathbf{f}\|_1} = \frac{\sum_i f_i y_i}{\sum_i f_i}$$

Alternatively, we could consider other measures of incorporating the scores such geometric mean or variance. One particularly interesting measure would be the smoothness along the set boundary,

$$\sum_{\{i|f_i=1\}} \sum_{\{j|f_j=0\}} W_{ij}(y_i - y_j)^2$$

This objective would, to some extent, detect local "patches" of relevant documents nested in some larger cluster.

Selecting and combining these objectives is not trivial. Although very similar to the isoperimeter problem,[2] the fact that our documents have scores associated with them makes the problem a more difficult boolean programming task. Semidefinite relaxations of this problem may provide good approximate solutions [Vandenberghe and Boyd, 1996; Poljak et al., 1995].

8.3.2 Incorporating Regularization into Formal Models

Local score regularization is presented as a fix for existing retrieval methods which ignore local consistency. One of the primary goals of this thesis is to prompt the introduction of

[2]The smallest-enclosing hypersphere problem for continuous spaces or isoperimetric set problem for graphs refer to the task of finding point sets of maximum support ([Scholkopf et al., 2001], [Chung et al., 2000], and [Grady and Schwartz, 2006] provide starting points for this literature).

regularity as a design principle for new retrieval systems and models. We have demonstrated that, for some retrieval scenarios, local consistency can significantly improve performance. We believe that the direct introduction of local consistency in formal models can result much stronger improvements. As discussed in Chapter 6, pseudo-relevance feedback captures this to a certain extent. Finding approaches which model local consistency using more than a single iteration of regularization is a compelling problem and a worthwhile research direction.

8.3.3 Cross-Media Regularization

In the same way we projected document scores across languages, we can consider projecting scores across media. Cross-media information retrieval refers to the scenario where the user poses a query in one media, for example text, and expects results in some other media, for example images. In *cross-media regularization*, we treat the collections in Figure 8.3 as coming from different media. However, we need to ask ourselves two questions. First, do we have a meaningful parallel corpus? The answer to this question depends on the task but in some situations, we can reply affirmatively. For example, images and movies often can be associated with explicit keywords or, when available, the text in which it is situated. Second, we must ask whether an appropriate affinity measure exists in the target corpus. Certainly we have presented substantial evidence that, for text, content-based similarity measures are appropriate for topic-based retrieval. It is less clear that we can make similar claims about other media. We have some evidence that appropriate similarity measures exist for some images but, in general, this is an open research question [He et al., 2004; Shi and Malik, 2000].

8.3.4 Diffusion Wavelets

We mentioned in passing that regularization using lower order harmonic functions on the graph did not improve performance as much as the local approach we take. Although we can argue that the peaked nature of the retrieval function precludes harmonic reconstruction, this does not imply that our local approach is necessarily the best approach. For example, multi-scale analysis and diffusion wavelets [Bremer et al., 2004; Coifman and Maggioni, 2004; Szlam et al., 2006] would provide a middle ground between regularization based on global analysis and regularization based purely on local analysis.

8.3.5 Robust Regularization

One of the assumptions underlying regularization is that all initial retrieval scores are equally valid. This is represented in the error cost,

$$\mathcal{E}(\mathbf{f}, \mathbf{y}) = \sum_{i=1}^{\tilde{n}} (f_i - y_i)^2$$

Retrieval algorithms rarely are equally confident about document scores. An system is often more confident about scores of high-scoring documents than low-scoring documents. Unfortunately, our constraint is ignorant of these subtleties. In reality what we would like is for

our constraint to penalize inconsistencies with high-scoring documents more than inconsistencies with low-scoring documents. One way to introduce this adaptive weight is to define a new error cost,

$$\mathcal{E}_g(\mathbf{f}, \mathbf{y}) = \sum_{i=1}^{\tilde{n}} g(y_i)(f_i - y_i)^2 \tag{8.12}$$

where g is a monotonically decreasing function of the rank of the document i; lower-ranked documents contribute less to the cost. This type of adaptive weighting would result in a regularization in which low-scoring documents related to high-scoring documents would bubble up the ranked list without allowing high-scoring documents to be weighed down by low-scoring neighbors.

CHAPTER 9

CONCLUSIONS

We began this thesis by describing the cluster hypothesis as a design principle for information retrieval systems. In the course of this dissertation, we have developed methods for measuring the satisfaction of this principle (Chapters 3 and 4), demonstrating that local consistency correlates positively with system performance. In Chapter 5, we moved from this correlation result to provide evidence of a causal relationship between local consistency and performance. Our technical contributions to information retrieval include

1. **A formal measure of local consistency.** We derived autocorrelation directly from the Voorhees test, allowing it to be used to measure the degree to which a system satisfies the cluster hypothesis.

2. **A demonstration of the correlation between local consistency and performance.** We presented empirical evidence that shows the relationship between local consistency and performance.

3. **A demonstration that improving the local score consistency of a system using a Laplacian-based approach improves performance.** We presented an algorithm which used the graph Laplacian to improve the local consistency of retrieval score functions. This demonstrated a causal relationship between local consistency and performance.

4. **A regularization-based perspective on pseudo-relevance feedback.** We presented an extended discussion of the relationship between regularization and previous research, concluding that some of the success of these methods may be explained by their effect on local score consistency.

These technical contributions all advance the understanding the cluster hypothesis in information retrieval.

In addition to these theoretical contributions, our work has resulted in several practical contributions to information retrieval,

1. **A novel precision prediction method directly.** We developed a new precision predictor which is competitive with state of the art precision predictors and improve performance when used in combination with these previous approaches.

2. **A consistently beneficial document re-ranking algorithm.** We described a new method based on the graph Laplacian for re-ranking documents based on improving local score consistency. We demonstrated that this algorithm is generally applicable and easily-extendable into new domains.

These two contributions have been rigorously tested across a diverse set of retrieval scenarios.

We believe this concluding chapter is the most appropriate place to nestle a few editorial comments about the local consistency and feature-based retrieval models. Feature-based models allow the designer to use sophisticated machine learning techniques to select parameter values which optimize performance for retrieval evaluation measures [Metzler and Croft, 2005; Yue et al., 2007]. The resulting complex combination of features often significantly improves performance. Unfortunately, to date, these approaches lack one fundamental property of term-based models: the ability to model local consistency. Similarity in term space tends to imply topical relatedness. Similarity in feature-space does not necessarily imply topical relatedness. If documents are only ever represented by abstract features, then computing topical relationships is very difficult. We believe that these models should formally and directly incorporate topical regularity objectives just as they do other features and hopefully in a manner more elaborate than a single iteration of regularization. The beauty of these models is in their ability to automatically combine well-known design principles from information retrieval. The cluster hypothesis should not be excluded.

That said, we should be explicit about the limitations of our work. First, the improvements garnered by regularization were most visible at higher recall points. If we are building a system for a high precision task such as web search, then we will only see marginal gains from regularization. However, there are many retrieval tasks for which all recall points are important; these include legal search and medical search. Second, regularization requires a meaningful affinity matrix, $\mathbf{W}$, where we expect labels or scores of related documents to be similar. There are certainly tasks where affinity measures are noisier (for example, sentence affinity) or not related to scores at all (for example, diversity-based ranking or novelty). Finally, some of the methods in this work are admittedly intended to obsolesce. We aim to provoke the incorporation of regularization terms into existing and future retrieval models. If this dissertation is successful, systems will produce locally consistent scores, preventing prediction by autocorrelation or improvement by *post-hoc* regularization.

APPENDIX A

EXPERIMENTAL DATA

A.1 Data for Detailed Experiments

A.1.1 Topics

We performed experiments on two data sets. The first data set, which we will call "trec12", consists of the 150 TREC Ad Hoc topics 51-200. We used only the news collections on Tipster disks 1 and 2 [Harman, 1993]. The second data set, which we will call "robust", consists of the 250 TREC 2004 Robust topics [Voorhees, 2004]. We used only the news collections on TREC disks 4 and 5. The robust topics are considered to be difficult and have been constructed to focus on topics which systems usually perform poorly on. For both data sets, we use only the topic title field as the query. The topic title is a short, keyword query associated with each TREC topic. We indexed collections using the Indri retrieval system, the Rainbow stop word list, and Krovetz stemming [Strohman et al., 2004; McCallum, 1996; Krovetz, 1993].

A.1.2 Baselines

For these detailed experiments, we sought baselines which were strong, in the sense of high performance, and realistic, in the sense of not over-fitting. Therefore, we first performed cross-validation to construct baseline retrieval scores. We report the specifics of these experiments in the subsequent sections. We describe our experimental data in Section A.1.1 and baseline algorithms in Section 2.2.1-2.2.3. We present parameters for our baseline algorithms in Table A.2. We also present trained parameter values (or ranges if they were different across partitions).[1]

[1]Our Markov random field baseline system uses a structured query model which incorporates inter-term dependencies [Metzler and Croft, 2005]. We use the Indri query language to implement full dependence models with fixed parameters of $(\lambda_T, \lambda_O, \lambda_U) = \{0.8, 0.1, 0.1\}$ as suggested by the authors [Metzler and

	number of documents	queries	comments
trec12	741,856	51-200	Tipster disks 1 and 2 without government documents
robust	472,525	301-450,650-700	TREC disks 4 and 5 without government documents

Table A.1. Topics and corpora used in detailed experiments.

	range	optimal trec12	optimal robust
Okapi			
b	[0.1-1.0; 0.1]	0.3	0.3
k	[0.5-2.5; 0.25]	1.5-2.0	0.75
Query Likelihood			
μ	[500-4000; 500]	2000	1000
Relevance Models			
r	{5, 25, 50, 100}	25-50	5-25
$\tilde{m}$	{5, 10, 25, 50, 75, 100}	100	75-100
λ	[0.1-0.7; 0.1]	0.2	0.1-0.2
Markov Random Field			
μ_{text}	[500-4000; 500]	500-1500	3000-4000
μ_{window}	[500-4000; 500]	500-2000	500

Table A.2. Parameter sweep values. Parameter ranges considered in the cross-validation. For each topic set, we present the optimal parameter values selected during training. When these values were not stable across partitions, we present the optimal parameter ranges.

A.2 Data for Generalizability Experiments

A.2.1 Collections and Runs

In addition to our detailed experiments, we were interested in evaluating the generalizability of score regularization to arbitrary initial retrieval algorithms. To this end, we collected the document rankings for all automatic runs submitted to the Ad Hoc Retrieval track for TRECs 3-8, Robust 2003-2005, Terabyte 2004-2005, TRECs 3-4 Spanish, and TRECs 5-6 Chinese [Voorhees and Harman, 2001]. This constitutes a variety of runs and tasks with varying levels of performance. In all cases, we use the appropriate evaluation corpora, not just the news portions as in the detailed experiments. We also include results for the TREC 2005 Enterprise track Entity Retrieval subtask. This subtask deals with the modeling and retrieval of entities mentioned in an enterprise corpus consisting of email and webpages. Although all sites participating in TREC include a score in run submissions, we cannot be confident about the accuracy of the scores. Therefore, inconsistent behavior for some runs may be the result of inaccurate scores. We present statistics for these collections and runs in Table A.3.

Croft, 2005]. We focus training for the Markov random field on the feature parameters governing smoothing. See [Metzler and Croft, 2005] for a more detailed description of these parameters.

	number of documents	queries	number of runs
trec3	741,856	151-200	28
trec4	567,529	201-250	14
trec5	524,929	251-300	30
trec6	556,077	301-350	56
trec7	528,155	351-400	86
trec8	528,155	401-450	116
robust03	528,155	601-650, robust03	76
robust04	528,155	301-450, 651-700	80
robust05	1,033,461	robust05	59
terabyte04	25,205,179	701-750	70
terabyte05	25,205,179	751-800	54
trec4-spanish	57,868	26-50	21
trec5-spanish	230,820	26-50	18
trec5-chinese	164,779	1-28	20
trec6-chinese	164,779	29-54	28
ent05	1,092	1-50	37

Table A.3. Topics, corpora, and runs used in generalizability experiments.

A.2.2 Affinity Matrices

We used the cosine similarity described in Section 2.3. Non-English collections received no linguistic processing: tokens were broken on whitespace for Spanish and single characters were used for Chinese. Entity similarity is determined by the cooccurence of entity names in the corpus.

APPENDIX B

SYMBOLS

$\mathbf{A}$	matrix
$\mathbf{A}_i$	the ith matrix
A_{ij}	element (i, j) of matrix $\mathbf{A}$
$\mathbf{a}$	vector
$\mathbf{a}_i$	the ith vector
a_i	element i of vector $\mathbf{a}$
a	scalar
$f(\mathbf{A})$	element-wise function of $\mathbf{A}$
$\mathbf{A}^{1/2}$	element-wise square root
$\mathbf{A}^{-1}$	matrix inverse
$\mathbf{A}^\mathsf{T}$	matrix transpose
$\|\|\mathbf{a}\|\|_i$	L_i norm of the vector $\mathbf{a}$

Table B.1. Notational convention for vector and matrix representation.

n	number of documents
$\tilde{n}$	number of documents to regularize
m	number of terms
$\mathbf{C}$	$n \times m$ collection matrix
$\mathbf{d}_i$	row i of $\mathbf{C}$ as a column vector
$\mathbf{w}_i$	column i of $\mathbf{C}$
$\mathbf{l}$	$n \times 1$ vector of document lengths
$\mathbf{c}$	$m \times 1$ vector of term document frequencies
$\mathbf{A}$	$n \times n$ document affinity matrix
$\mathbf{W}$	nearest neighbor graph based on $\mathbf{A}$
$\mathbf{y}$	$n \times 1$ initial score vector
$\mathbf{f}$	$n \times 1$ regularized score vector
$\mathbf{U}$	$m \times k$ matrix of cluster vectors
$\mathbf{V}$	$k \times n$ matrix of documents embedded into k dimensions
$\mathbf{y}_c$	$k \times 1$ cluster score vector
$\mathbf{W}_e$	$n \times n$ graph based on expanded documents
$\mathbf{y}_e$	$n \times 1$ vector of scores for expanded documents
Δ	$n \times n$ Laplacian on $\mathbf{W}$
$\mathbf{E}_k$	$n \times k$ matrix of top k eigenvectors of $\mathbf{W}$
$\mathbf{e}$	column vector of all 1's
$\mathbf{I}$	identity matrix

Table B.2. Definition of Symbols.

APPENDIX C

EVALUATION

C.1 Metrics

Although the goal of information retrieval is a classification of all documents as relevant or non-relevant, the highly skewed class distribution requires the adoption of rank-based measures. In all experiments we will be measuring performance using *mean average precision* and *interpolated precision at standard recall levels*. Following convention, we will be selecting parameters to optimize mean average precision.

For all experiments, we evaluate the top 1000 documents retrieved. Let this ranked set of documents be defined as the vector ρ_q for a particular query, q, such that $\rho_q[i]$ is the relevance judgment of the document at rank i. That is, $\rho_q[i] = 1$ if the ith ranked document is relevant and $\rho_q[i] = 0$ otherwise. We often want to evaluate the quality of the ranking after a user has observed k documents in the ranking. Precision after k documents is defined as,

$$\mathcal{P}_k(\rho_q) = \frac{1}{k}\sum_{i=1}^{k}\rho_q[i] \tag{C.1}$$

Recall after k documents is defined as,

$$\mathcal{R}_k(\rho_q) = \frac{1}{|R_q|}\sum_{i=1}^{k}\rho_q[i] \tag{C.2}$$

where $|R_q|$ is the number of relevant documents for the query.

C.1.1 Mean Average Precision

The average precision for a query is defined as,

$$\overline{\mathcal{P}}(\rho_q) = \frac{1}{|R_q|}\sum_{k=1}^{N}\mathcal{P}_k(\rho_q)\times\rho_q[k] \tag{C.3}$$

where N is the total number of documents retrieved. We can combine the average precision for a set of queries by using the arithmetic mean,

$$\overline{\mathcal{P}} = \frac{1}{|Q|}\sum_{q\in Q}\overline{\mathcal{P}}(\rho_q) \tag{C.4}$$

where Q is our set of queries. We refer to this as the mean average precision (map) of a particular algorithm for a set of queries.

C.1.2 Interpolated Precision at Standard Recall Levels

Although favoring systems which place relevant documents at the beginning of the ranked list, the mean average precision does not provide a good indication of performance at particular recall levels. We would like to to say, for example, that a particular method demonstrates high precision near the top of the ranked list as opposed to closer to the bottom.

A finer-grained method for measuring ranked list performance is to use the interpolated precision at specific recall levels. The precision at a recall level x refers to the precision value of $\mathcal{P}_k(\rho_q)$ where $\mathcal{R}_k(\rho_q) = x$. For a particular query, the recall function jumps in increments of $\frac{1}{|R_q|}$. Therefore, we must interpolate the precision using the sampled recall points. It is common practice in information retrieval to define the interpolated precision at recall level x as,

$$\widetilde{\mathcal{P}}_x(\rho_q) \quad = \quad \sup\{\mathcal{P}_k(\rho_q) : \mathcal{R}_k(\rho_q) \geq x\} \tag{C.5}$$

In practice, this results in a monotonically decreasing step function. We present the interpolated precision graphically as the colored functions in Figure C.1. The interpolated precision at recall level x for a set of queries averages these values,

$$\widetilde{\mathcal{P}}_x \quad = \quad \frac{1}{|Q|} \sum_{q \in Q} \widetilde{\mathcal{P}}_x(\rho_q) \tag{C.6}$$

where Q is our set of queries. This is shown graphically in Figure C.1 as the black line. We use the convention of computing $\widetilde{\mathcal{P}}_x$ at the recall levels,

$$\{0.00, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, 1.00\}.$$

C.2 Cross Validation

Whenever parameters needed tuning, we performed 10-fold cross-validation. We adopt a Platt's cross-validation evaluation for training and evaluation [Platt, 2000]. We first randomly partition the queries for a particular collection. For each partition, i, the algorithm is trained on all but that partition and is evaluated using that partition, i. For example, if the training phase considers the topics and judgments in partitions 1-9, then the testing phase uses the optimal parameters for partitions 1-9 to perform retrieval using the topics in partition 10. Using each of the ten possible training sets of size nine, we generate unique evaluation rankings for each of the topics over all partitions. Evaluation and comparison was performed using the union of these ranked lists.

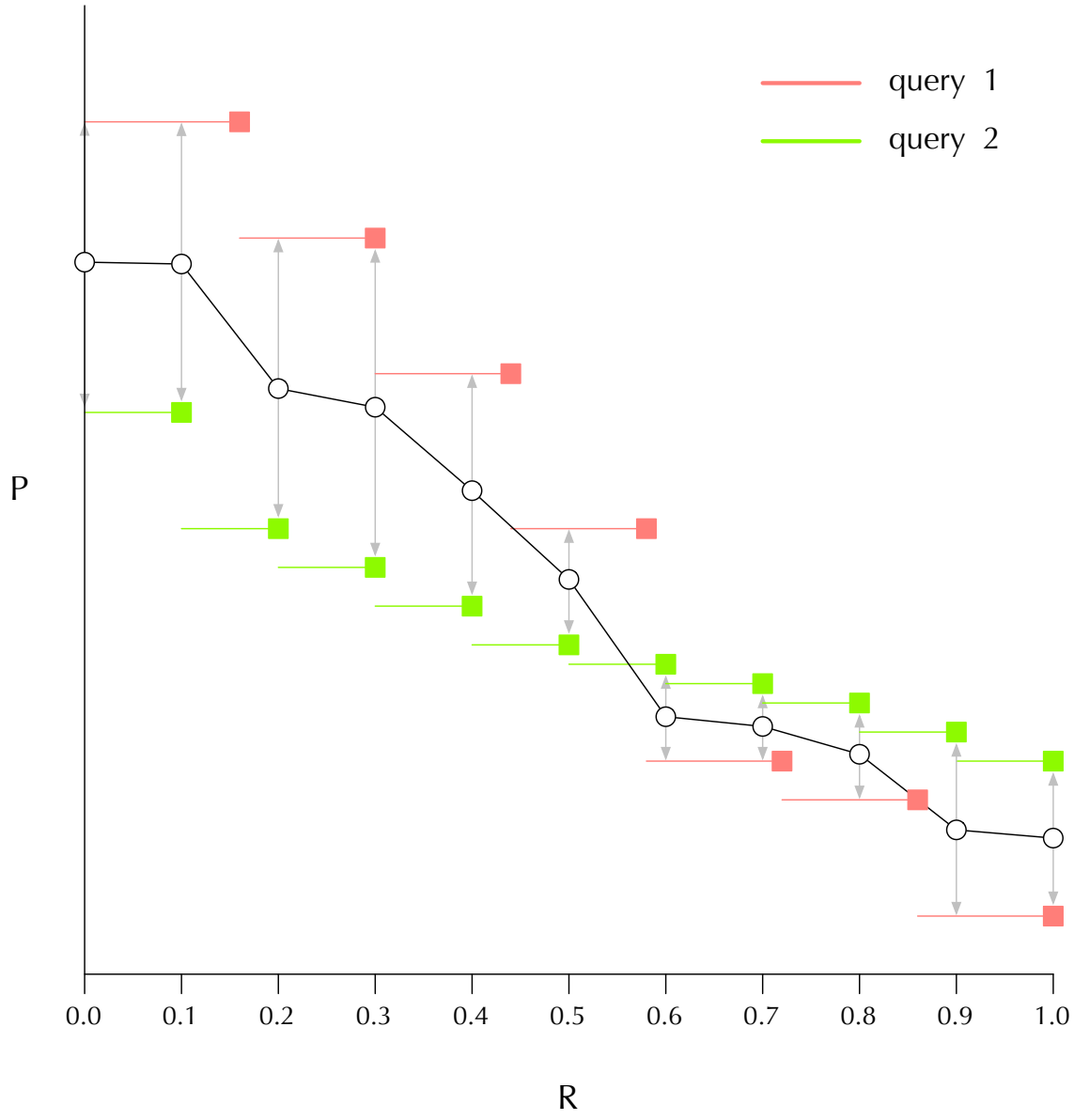


Figure C.1. Averaging interpolated precision curves. The interpolated precision curves for two queries are shown in color. The average interpolated precision can be computed at standard recall levels and is depicted as the solid black line.

APPENDIX D

DETAILED RESULTS

trec12/okapi						
	0	100	250	500	750	1000
0.00	0.7430	0.7508	0.7521	0.7473	0.7485	0.7576
0.10	0.5086	**0.5188**	**0.5180**	**0.5211**	**0.5204**	**0.5245**
0.20	0.4308	**0.4367**	**0.4483**	**0.4556**	**0.4553**	**0.4582**
0.30	0.3693	0.3725	**0.3862**	**0.3964**	**0.3997**	**0.4049**
0.40	0.3045	0.3095	**0.3215**	**0.3331**	**0.3436**	**0.3462**
0.50	0.2538	0.2560	**0.2620**	**0.2716**	**0.2804**	**0.2865**
0.60	0.2015	0.2022	**0.2062**	**0.2185**	**0.2252**	**0.2308**
0.70	0.1420	0.1428	**0.1458**	**0.1524**	**0.1600**	**0.1653**
0.80	0.0897	0.0908	**0.0947**	**0.0981**	**0.1027**	**0.1047**
0.90	0.0396	0.0395	0.0430	0.0438	**0.0449**	**0.0473**
1.00	0.0042	0.0042	0.0042	0.0052	0.0052	0.0046
map	0.2600	**0.2632**	**0.2693**	**0.2754**	**0.2788**	**0.2834**

Table D.1. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing Okapi scores for trec12 collection as a function of the number of regularized documents. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

robust/okapi						
	0	100	250	500	750	1000
0.00	0.7361	0.7311	0.7232	0.7175	*0.7030*	*0.7050*
0.10	0.5388	0.5441	**0.5600**	**0.5606**	**0.5497**	**0.5550**
0.20	0.4346	**0.4512**	**0.4641**	**0.4589**	**0.4546**	**0.4600**
0.30	0.3667	0.3734	**0.3890**	**0.3874**	**0.3868**	**0.3887**
0.40	0.2913	**0.3012**	**0.3153**	**0.3170**	**0.3128**	**0.3181**
0.50	0.2353	**0.2507**	**0.2615**	**0.2669**	**0.2615**	**0.2727**
0.60	0.1894	**0.1987**	**0.2088**	**0.2162**	**0.2131**	**0.2226**
0.70	0.1538	**0.1563**	**0.1637**	**0.1692**	**0.1697**	**0.1713**
0.80	0.1059	0.1057	0.1106	**0.1167**	**0.1202**	**0.1185**
0.90	0.0666	0.0683	**0.0719**	**0.0759**	**0.0774**	**0.0772**
1.00	0.0338	0.0351	**0.0369**	**0.0379**	**0.0376**	**0.0374**
map	0.2652	**0.2713**	**0.2804**	**0.2827**	**0.2791**	**0.2826**

Table D.2. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing Okapi scores for robust collection as a function of the number of regularized documents. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

trec12/QL						
	0	100	250	500	750	1000
0.00	0.7518	0.7480	0.7462	0.7501	0.7451	0.7330
0.10	0.4922	**0.5033**	**0.5211**	**0.5272**	**0.5274**	**0.5244**
0.20	0.4163	**0.4266**	**0.4494**	**0.4593**	**0.4620**	**0.4626**
0.30	0.3469	**0.3545**	**0.3794**	**0.3909**	**0.3993**	**0.4014**
0.40	0.2913	**0.2968**	**0.3178**	**0.3303**	**0.3402**	**0.3460**
0.50	0.2325	**0.2387**	**0.2501**	**0.2658**	**0.2742**	**0.2818**
0.60	0.1792	0.1827	**0.1910**	**0.2048**	**0.2119**	**0.2142**
0.70	0.1345	0.1353	**0.1401**	**0.1490**	**0.1557**	**0.1566**
0.80	0.0953	0.0958	**0.0991**	0.0990	**0.1081**	**0.1104**
0.90	0.0493	0.0490	0.0502	0.0494	0.0524	0.0532
1.00	0.0084	0.0084	0.0078	0.0060	0.0055	0.0057
map	0.2506	**0.2554**	**0.2657**	**0.2722**	**0.2783**	**0.2800**

Table D.3. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing QL scores as a function of the number of regularized documents for the trec12 collection. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

robust/QL						
	0	100	250	500	750	1000
0.00	0.7523	*0.7398*	0.7402	*0.7317*	*0.7270*	0.7348
0.10	0.5420	**0.5633**	**0.5652**	**0.5642**	**0.5631**	**0.5692**
0.20	0.4375	**0.4622**	**0.4713**	**0.4711**	**0.4715**	**0.4763**
0.30	0.3605	**0.3872**	**0.4028**	**0.4038**	**0.4091**	**0.4063**
0.40	0.2843	**0.3131**	**0.3281**	**0.3340**	**0.3377**	**0.3411**
0.50	0.2356	**0.2600**	**0.2741**	**0.2828**	**0.2883**	**0.2882**
0.60	0.1880	**0.2013**	**0.2165**	**0.2188**	**0.2243**	**0.2295**
0.70	0.1477	0.1533	**0.1657**	**0.1692**	**0.1739**	**0.1736**
0.80	0.1040	0.1038	**0.1124**	**0.1206**	**0.1197**	**0.1199**
0.90	0.0696	0.0732	**0.0763**	**0.0769**	**0.0786**	**0.0804**
1.00	0.0398	0.0427	0.0417	0.0411	0.0403	0.0405
map	0.2649	**0.2788**	**0.2885**	**0.2909**	**0.2933**	**0.2955**

Table D.4. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing QL scores as a function of the number of regularized documents for the robust collection. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

trec12/RM						
	0	100	250	500	750	1000
0.00	0.7766	0.7645	0.7754	0.7602	0.7673	0.7740
0.10	0.5489	**0.5623**	**0.5609**	0.5600	0.5613	0.5575
0.20	0.4882	0.4911	0.4919	0.4940	0.4971	0.4946
0.30	0.4350	0.4392	**0.4411**	**0.4452**	**0.4449**	**0.4453**
0.40	0.3797	0.3819	**0.3894**	**0.3945**	**0.3947**	**0.3929**
0.50	0.3210	0.3243	**0.3329**	**0.3432**	**0.3437**	**0.3426**
0.60	0.2666	0.2683	**0.2736**	**0.2844**	**0.2873**	**0.2865**
0.70	0.2017	0.2053	**0.2089**	**0.2171**	**0.2187**	**0.2171**
0.80	0.1424	0.1407	0.1432	**0.1491**	**0.1543**	**0.1548**
0.90	0.0865	0.0860	0.0871	0.0859	0.0877	0.0871
1.00	0.0098	0.0098	0.0097	0.0095	0.0094	0.0102
map	0.3154	0.3176	**0.3203**	**0.3248**	**0.3257**	**0.3252**

Table D.5. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing RM scores as a function of the number of regularized documents for the trec12 collection. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

robust/RM						
	0	100	250	500	750	1000
0.00	0.6926	0.7005	*0.6879*	0.6909	0.6931	0.6904
0.10	0.5458	**0.5593**	0.5531	0.5533	0.5552	0.5521
0.20	0.4691	**0.4842**	**0.4811**	**0.4830**	**0.4826**	**0.4817**
0.30	0.4030	**0.4168**	**0.4135**	**0.4176**	**0.4189**	**0.4178**
0.40	0.3410	**0.3536**	**0.3571**	**0.3620**	**0.3642**	**0.3613**
0.50	0.2918	**0.3036**	**0.3062**	**0.3072**	**0.3083**	**0.3088**
0.60	0.2410	0.2473	**0.2494**	**0.2559**	**0.2553**	**0.2543**
0.70	0.1915	0.1970	**0.2016**	**0.2053**	**0.2052**	**0.2001**
0.80	0.1445	**0.1535**	**0.1577**	**0.1547**	**0.1573**	**0.1546**
0.90	0.0901	**0.0967**	0.0981	**0.1009**	**0.1007**	**0.1002**
1.00	0.0448	0.0450	0.0450	0.0476	0.0499	**0.0504**
map	0.2961	**0.3051**	**0.3041**	**0.3068**	**0.3084**	**0.3058**

Table D.6. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing RM scores as a function of the number of regularized documents for the robust collection. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

trec12/MRF						
	0	100	250	500	750	1000
0.00	0.7440	**0.7616**	**0.7540**	**0.7513**	0.7401	0.7335
0.10	0.5091	**0.5197**	**0.5250**	**0.5323**	**0.5316**	**0.5319**
0.20	0.4270	**0.4399**	**0.4536**	**0.4691**	**0.4702**	**0.4712**
0.30	0.3565	**0.3642**	**0.3831**	**0.4006**	**0.4113**	**0.4140**
0.40	0.3029	0.3080	**0.3241**	**0.3431**	**0.3553**	**0.3565**
0.50	0.2465	**0.2532**	**0.2619**	**0.2819**	**0.2885**	**0.2974**
0.60	0.1975	0.1994	**0.2101**	**0.2220**	**0.2287**	**0.2291**
0.70	0.1444	0.1488	**0.1509**	**0.1554**	**0.1610**	**0.1640**
0.80	0.1014	0.1021	0.1062	0.1062	**0.1099**	**0.1103**
0.90	0.0525	0.0523	0.0526	0.0514	0.0523	0.0531
1.00	0.0084	0.0084	0.0078	0.0061	0.0056	0.0056
map	0.2603	**0.2657**	**0.2724**	**0.2810**	**0.2857**	**0.2874**

Table D.7. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing Markov random field scores for trec12 collection as a function of the number of regularized documents. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

	robust/MRF					
	0	100	250	500	750	1000
0.00	0.7601	0.7561	*0.7495*	0.7522	*0.7418*	*0.7442*
0.10	0.5640	**0.5787**	**0.5853**	**0.5870**	**0.5833**	**0.5857**
0.20	0.4486	**0.4665**	**0.4799**	**0.4891**	**0.4820**	**0.4826**
0.30	0.3762	**0.3933**	**0.4115**	**0.4208**	**0.4135**	**0.4152**
0.40	0.3020	**0.3209**	**0.3331**	**0.3516**	**0.3477**	**0.3487**
0.50	0.2508	**0.2660**	**0.2855**	**0.2957**	**0.3002**	**0.3005**
0.60	0.1992	**0.2055**	**0.2289**	**0.2368**	**0.2394**	**0.2402**
0.70	0.1586	0.1594	**0.1725**	**0.1787**	**0.1802**	**0.1800**
0.80	0.1081	0.1102	**0.1169**	**0.1311**	**0.1295**	**0.1298**
0.90	0.0775	0.0801	**0.0829**	**0.0837**	**0.0822**	**0.0817**
1.00	0.0419	0.0451	0.0459	**0.0457**	0.0447	0.0434
map	0.2769	**0.2856**	**0.2966**	**0.3038**	**0.3025**	**0.3030**

Table D.8. Average interpolated precision at standard recall points and mean average non-interpolated precision. This table demonstrates performance of regularizing Markov random field scores for robust collection as a function of the number of regularized documents. Bold numbers indicate statistically significant improvements in performance using the Wilcoxon test ($p < 0.05$); italicized numbers indicate statistically significant degradations in performance.

BIBLIOGRAPHY

IJsbrand Jan Aalbersberg. Incremental relevance feedback. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–22, ACM Press, 1992.

Alex T. Adai, Shailesh V. Date, Shannon Wieland, and Edward M. Marcotte. Lgl: Creating a map of protein function with an algorithm for visualizing very large biological networks. *Journal of Molecular Biology*, 340(1):179–190, 2004.

James Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*, volume 12 of *The Information Retrieval Series*. Springer, New York, NY, USA, 2002.

Javed Aslam and Virgil Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *ECIR 2007: Proceedings of the 29th European Conference on Information Retrieval*, 2007.

Jaroslaw Baliński and Czeslaw Daniłowicz. Re-ranking method based on inter-document distances. *Inf. Process. Manage.*, 41(4):759–775, 2005.

Ron Bekkerman and James Allan. Using bigrams in text categorization. Technical Report IR-408, Center for Intelligent Information Retrieval, 2004.

R. K. Belew. Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. In *SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–20, ACM Press, 1989.

Mikhail Belkin and Partha Niyogi. Using manifold stucture for partially labeled classification. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 929–936. MIT Press, Cambridge, MA, 2003.

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, pages 624–638, 2004.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

James C Bremer, Ronald Coifman, Mauro Maggioni, and Arthur D Szlam. Diffusion wavelet packets. Technical Report YALE/DCS/TR-1304, Yale University, 2004.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Elsevier Science Publishers B. V., 1998.

Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 33–40, ACM Press, 2000.

David Carmel, Elad Yom-Tov, Adam Darlow, and Dan Pelleg. What makes a query difficult? In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 390–397, ACM Press, 2006.

Ben Carterette and Desislava Petkova. Learning a ranking from pairwise preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 629–630, ACM Press, 2006.

Francine R. Chen, Ayman O. Farahat, and Thorsten Brant. Multiple similarity measures and source-pair information in story link detection. In *Human Language Technology Conference, North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004)*, pages 313–320, May 2004.

Zhe Chen and Simon Haykin. On different facets of regularization theory. *Neural Comput.*, 14(12):2791–2846, 2002.

Fan Chung. Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9:1–19, 2004.

Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

Fan R. K. Chung, Alexander Grigor'yan, and Shing-Tung Yau. Higher eigenvalues and isoperimetric inequalities on riemannian manifolds and graphs. *Communications in Analysis and Geometry*, 8(5):969–1026, 2000.

A. D. Cliff and J. K. Ord. *Spatial Autocorrelation*. Pion Ltd., 1973.

David A. Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *NIPS*, pages 430–436, 2000.

Ronald Coifman and Mauro Maggioni. Diffusion wavelets. Technical Report YALE/DCS/TR-1303, Yale University, 2004.

Ronald R Coifman and Stephane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006.

T. Cover. Estimation by the nearest neighbor rule. *Information Theory, IEEE Transactions on*, 14(1):50–55, 1968.

F. Crestani. Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.*, 11(6):453–482, 1997.

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

W. B. Croft and D. J. Harper. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation*, 35(4):285–295, 1979.

W. B. Croft and H. Turtle. A retrieval model incorporating hypertext links. In *HYPERTEXT '89: Proceedings of the second annual ACM conference on Hypertext*, pages 213–224, ACM Press, 1989.

W. B. Croft, T. J. Lucia, and P. R. Cohen. Retrieving documents by plausible inference: a preliminary study. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 481–494, ACM Press, 1988.

W. Bruce Croft and John Lafferty. *Language Modeling for Information Retrieval*. Kluwer Academic Publishing, 2003.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, ACM Press, 2002.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. Precision prediction based on ranked list coherence. *Information Retrieval*, 9(6):723–755, 2006.

Wei Dai and Rohini Srihari. Minimal document set retrieval. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 752–759, ACM Press, 2005.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

Luc Devroye. The uniform convergence of nearest neighbor regression function estimators and their application in optimization. *IEEE Transactions on Information Theory*, 24(2): 142–151, March 1978.

Fernando Diaz and Rosie Jones. Using temporal profiles of queries for precision prediction. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–24, ACM Press, 2004.

Fernando Diaz and Donald Metzler. Pseudo-aligned multilingual corpora. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2727–2732, 2007.

Mark D. Dunlop. The effect of accessing nonmatching documents on relevance feedback. *ACM Trans. Inf. Syst.*, 15(2):137–153, 1997.

Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, ACM Press, 2004.

Christian Genest and Jean-Francois Plante. On Blest's measure of rank correlation. *The Canadian Journal of Statistics*, 31(1):1–18, 2003.

Leo Grady and Eric L. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):469–475, 2006. Member-Leo Grady.

Daniel A. Griffith. *Spatial Autocorrelation and Spatial Filtering*. Springer Verlag, 2003.

Alan Griffiths, H. Clair Luckhurst, and Peter Willett. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science*, 37(1):3–11, 1986.

Donna K. Harman. The first text retrieval conference (TREC-1) Rockville, MD, U.S.A., 4-6 November, 1992. *Information Processing and Management*, 29(4):411–414, 1993.

B. He and I. Ounis. Inferring Query Performance Using Pre-retrieval Predictors. In *The Eleventh Symposium on String Processing and Information Retrieval (SPIRE)*, 2004.

Xiaofei He, Wei-Ying Ma, and Hong-Jiang Zhang. Learning an image manifold for retrieval. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 17–23, ACM Press, 2004.

Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84, ACM Press, 1996.

Desmond J. Higham. Condition numbers and their condition numbers. *Linear Algebra and Its Applications*, 214:193–213, January 1995.

Thomas Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, ACM Press, 1999.

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

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pages 668–677, Society for Industrial and Applied Mathematics, 1998.

Robert Krovetz. Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202, ACM Press, 1993.

Oren Kurland. *Inter-document Similarities, Language Models, and Ad Hoc Information Retrieval*. PhD thesis, Cornell University, Ithaca, NY, August 2006.

Oren Kurland and Lillian Lee. Corpus structure, language models, and ad hoc information retrieval. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 194–201, ACM Press, 2004.

Oren Kurland and Lillian Lee. Pagerank without hyperlinks: Structural re-ranking using links induced by language models. In *SIGIR '05: Proceedings of the 28th annual international conference on Research and development in information retrieval*, 2005.

Oren Kurland, Lillian Lee, and Carmel Domshlak. Better than the real thing?: iterative pseudo-query processing using cluster-based language models. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, ACM Press, 2005.

K. L. Kwok. A neural network for probabilistic information retrieval. In *SIGIR '89: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–30, ACM Press, 1989.

John Lafferty and Guy Lebanon. Diffusion kernels on statistical manifolds. *Journal of Machine Learning Research*, 6:129–163, 2005.

Stephane Lafon. *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, 2004.

G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies. i. hierarchical systems. *Computer Journal*, 9:373–380, 1967.

Victor Lavrenko. *A Generative Theory of Relevance*. PhD thesis, University of Massachusetts, 2004.

Victor Lavrenko and James Allan. Real-time query expansion in relevance models. Technical Report IR-473, University of Massachusetts Amherst, 2006.

Anton Leouski and James Allan. Visual interactions with a multidimensional ranked list. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 353–354, ACM Press, 1998.

Anton Leuski. *Interactive Information Organization: Techniques and Evaluation*. PhD thesis, University of Massachusetts Amherst, 2001.

Xiaoyan Li. Robust relevance-based language models. In *Proceedings of the Fourth IASTED International Conference on Communications, Internet and Information Technology (CIIT 2006)*, pages 341–348, 2006.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, January 1991.

Xiaoyong Liu and W. Bruce Croft. Cluster-based retrieval using language models. In *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 186–193, ACM Press, 2004.

Xiaoyong Liu and W. Bruce Croft. Experiments on retrieval of optimal clusters. Technical Report IR-478, University of Massachusetts Amherst, 2006.

R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining the outputs of search engines. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 267–275, ACM Press, 2001.

Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

Donald Metzler and W. Bruce Croft. A Markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, ACM Press, 2005.

Donald Metzler and W. Bruce Croft. Latent concept expansion using markov random fields. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2007)*, 2007.

Donald Metzler and W. Bruce Croft. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40(5):735–750, 2004.

Mark Montague and Javed A. Aslam. Relevance score normalization for metasearch. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 427–433, ACM Press, 2001.

Ramesh Nallapati and James Allan. Capturing term dependencies using a language model based on sentence trees. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 383–390, ACM Press, 2002.

A. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*, 2002.

Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Stable algorithms for link analysis. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–266, ACM Press, 2001.

Paul Ogilvie. Nearest neighbor smoothing of language models in IR. `http://www.cs.cmu.edu/~pto/courses/11-743/`, 2003.

Paul Over. Trec-5 interactive track report. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of the 5th Text REtrieval Conference (TREC-5)*, pages 29–56, 1996.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Peter J. Bartlett, Bernhard Schölkopf, Dale Schuurmans, and Alex J. Smola, editors, *Advances in Large Margin Classifiers*. MIT Press, 2000.

S. Poljak, F. Rendl, and H. Wolkowicz. A recipe for semidefinite relaxation for $(0,1)$-quadratic programming. *Journal of Global Optimization*, 7:51–73, 1995.

Scott Preece. *A spreading activation network model for information retrieval*. PhD thesis, University of Illinois, Urbana-Champaign, 1981.

Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, Zheng Chen, and Wei-Ying Ma. A study of relevance propagation for web search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 408–415, ACM Press, 2005.

Matthew Richardson, Amit Prakash, and Eric Brill. Beyond pagerank: machine learning for static ranking. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 707–715, ACM Press, 2006.

S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 232–241, Springer-Verlag New York, Inc., 1994.

J. J. Rocchio. *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance Feedback in Information Retrieval, pages 313–323. Prentice-Hall Inc., 1971.

G. Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1971.

G. Salton and C. Buckley. On the use of spreading activation methods in automatic information. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 147–160, ACM Press, 1988.

Gerard. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill Text, 1968.

Jacques Savoy. Ranking schemes in hybrid boolean systems: a new approach. *J. Am. Soc. Inf. Sci.*, 48(3):235–253, 1997.

Bernhard Scholkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comp.*, 13(7):1443–1471, 2001.

Azadeh Shakery and ChengXiang Zhai. A probabilistic relevance propagation model for hypertext retrieval. In *Conference on Information and Knowledge Management*, 2006.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

Amit Singhal and Fernando Pereira. Document expansion for speech retrieval. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 34–41, ACM Press, 1999.

Alan F. Smeaton. Spanish and chinese document retrieval in TREC-5. In *TREC*, 1996.

G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

Trevor Strohman, Donald Metzler, Howard Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. In *Proceedings of the International Conference on Intelligence Analysis*, 2004.

Arthur D Szlam, Mauro Maggioni, and Ronald R Coifman. A general framework for adaptive regularization based on diffusion processes on graphs. Technical Report YALE/DCS/TR1365, Yale University, 2006.

Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. Language model information retrieval with document expansion. In *HLT/NAACL 2006*, pages 407–414, 2006.

H. Turtle and W. B. Croft. Inference networks for document retrieval. In *SIGIR '90: Proceedings of the 13th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 1–24, ACM Press, 1990.

C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

C. J. van Rijsbergen and Karen Sparck Jones. A test for the separation of relevant and non-relevant documents in experimental retrieval collections. *Journal of Documentation*, 29(3): 251–257, September 1973.

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Rev.*, 38(1): 49–95, 1996.

Vishwa Vinay, Ingemar J. Cox, Natasa Milic-Frayling, and Ken Wood. On ranking the effectiveness of searches. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 398–404, ACM Press, 2006.

Ellen M. Voorhees. The cluster hypothesis revisited. In *SIGIR '85: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 188–196, ACM Press, 1985.

Ellen M. Voorhees and Donna K. Harman, editors. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, 2001.

E.M. Voorhees. Overview of the TREC 2004 robust track. In *Proceedings of the 13th Text REtrieval Conference (TREC 2004)*, 2004.

Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, ACM Press, 2006.

Ross Wilkinson and Philip Hingston. Using the cosine measure in a neural network for document retrieval. In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 202–210, ACM Press, 1991.

Jinxi Xu and W. Bruce Croft. Cluster-based language models for distributed retrieval. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 254–261, ACM Press, 1999.

Elad Yom-Tov, Shai Fine, David Carmel, and Adam Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 512–519, ACM Press, 2005.

Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 271–278, ACM Press, 2007.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In L. Saul Thrun, S. and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, volume 16, pages 169–176, MIT Press, 2004.

Dengyong Zhou, Bernhard Schölkopf, and Thomas Hofmann. Semi-supervised learning on directed graphs. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1633–1640. MIT Press, Cambridge, MA, 2005.

Yun Zhou and W. Bruce Croft. Document quality models for web ad hoc retrieval. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 331–332, ACM Press, 2005.

Yun Zhou and W. Bruce Croft. Ranking robustness: a novel framework to predict query performance. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 567–574, ACM Press, 2006.