ANALYSIS OF THE FREQUENCY DISTRIBUTIONS OF QUERY TERMS ON DOCUMENT COLLECTIONS & PER-QUERY SELECTION OF BEST TERM-WEIGHTING MODEL

A dissertation submitted for the degree of
Doctor of Philosophy

Ahmet ARSLAN
Eskişehir, 2016

ANALYSIS OF THE FREQUENCY DISTRIBUTIONS OF QUERY TERMS ON DOCUMENT COLLECTIONS & PER-QUERY SELECTION OF BEST TERM-WEIGHTING MODEL

Ahmet ARSLAN

A dissertation submitted for the degree of
Doctor of Philosophy

Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Bekir Taner DİNÇER

Eskişehir
Anadolu University
Graduate School of Science
August, 2016

FINAL APPROVAL FOR THESIS

This thesis titled "**Analysis of the Frequency Distributions of Query Terms on Document Collections & Per-Query Selection of Best Term-Weighting Model**" has been prepared and submitted by **Ahmet Arslan** in partial fullfillment of the requirements in "Anadolu University Directive on Graduate Education and Examination" for the Degree of PhD in Computer Engineering Department has been examined and approved on 19/08/2016.

Committee Members		**Signature**
Member (Supervisor)	: Assoc. Prof. Dr. Bekir Taner DİNÇER	.....
Member	: Prof. Dr. Ümit Deniz TURAN	.....
Member	: Assoc. Prof. Dr. Cihan KALELİ	.....
Member	: Assist. Prof. Dr. Kemal ÖZKAN	.....
Member	: Assist. Prof. Dr. Mehmet KOÇ	.....

Director
Graduate School of Science

ABSTRACT

ANALYSIS OF THE FREQUENCY DISTRIBUTIONS OF QUERY TERMS ON DOCUMENT COLLECTIONS & PER-QUERY SELECTION OF BEST TERM-WEIGHTING MODEL

Ahmet ARSLAN

Department of Computer Engineering
Anadolu University, Graduate School of Science, August, 2016

Supervisor: Assoc. Prof. Dr. Bekir Taner DİNÇER

Many term-weighting models have been proposed for information retrieval but the effectiveness of each term-weighting model varies across queries (i.e., information needs of users). Thus, using a single term-weighting model to process all kinds of queries may not be appropriate for fulfilling every information need of users. Instead of using a single term weighting model, it is an empirical fact that using different term weighting models for different queries could provide an increase in information retrieval effectiveness by an order of magnitude. However, for any given query, automatically selecting the term-weighting model that could provide the highest achievable retrieval effectiveness in the current state-of-the-art of information retrieval technology is still an open and challenging research problem. This issue is, in general, referred to as *selective term weighting* or *selective weighting function* or *selective retrieval model* in the field of selective information retrieval. In this PhD dissertation, we will investigate a novel statistical/probabilistic approach to the *selective term weighting* problem, based on the frequency distributions of query terms on document collections.

A term-weighting model that works well for one query, may not work well for another. We are not capable of determining or justifying in advance the best term-weighting model to use with a given query. We know little of the characteristics of queries and document collections that affect the effectiveness of term-weighting models. This PhD dissertation aims to shed some light on this mystery by analyzing the frequency distributions of query terms on document collections.

All the results presented in this dissertation are fully repeatable and reproducible with data and code available online.

Keywords: Chi-Square Goodness-of-Fit Test, Index Term Weighting, Frequency Distribution, Robustness of Retrieval Effectiveness, Selective Information Retrieval.

ÖZET

BELGE DERLEMLERİNDE SORGU TERİMLERİNİN FREKANS DAĞILIMLARININ ANALİZİ VE SORGUYA GÖRE EN UYGUN TERİM AĞIRLIKLANDIRMA MODELİNİN SEÇİMİ

Ahmet ARSLAN

Bilgisayar Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Ağustos, 2016

Danışman: Doç. Dr. Bekir Taner DİNÇER

Bilgi erişimi için bir çok terim ağırlıklandırma modeli geliştirilmiştir. Fakat her terim ağırlıklandırma modelinin başarımı bazı sorgularda yüksek bazı sorgularda da düşüktür — başarımın gürbüzlüğü problemi. Diğer taraftan bir terim ağırlıklandırma modelinin başarımının düşük olduğu bir sorgu için diğer terim ağırlıklandırma modellerinin başarımı da düşük olmak zorunda değildir: herhangi bir sorgu için tatminkar düzeyde başarım sağlayacak bir terim ağırlıklandırma modelini mevcut teknolojiler içinde bulmak mümkün olabilir. Yani sisteme gelen her sorguyu tek bir terim ağırlıklandırma modeli ile cevaplamak, kullanıcıların bilgi ihtiyaçlarını en tatminkar şekilde karşılamak için uygun olmayabilir. Tüm sorgular için tekil bir terim ağırlıklandırma modeli kullanmak yerine, her bir ayrı sorgu için uygun bir terim ağırlıklandırma modeli kullanıldığında bilgi erişim başarımının mertebe kertesinde artış olduğu deneysel bir gerçektir. Ancak, verilen herhangi bir sorgu için en iyi başarımı sağlayacak olan modelin, bugünkü bilinen en gelişkin modeller arasından otomatik olarak seçiminin yapılması işi halen çözülememiş zor bir araştırma konusudur. Bu uğraş, seçkili bilgi erişimi çalışma alanında, genel olarak, seçkili terim ağırlıklandırma ya da seçkili ağırlıklandırma fonksiyonu olarak adlandırılır. Bu doktora tezinde, seçkili terim ağırlıklandırma uğraşı için sorgu terimlerinin derlemler üzerindeki frekans dağılımlarına dayanan özgün bir istatistik/olasılık esasında yaklaşım incelenmiştir.

Bir sorguda iyi çalışan terim ağırlıklandırma modeli başka bir sorguda iyi çalışmayabilmektedir. Verilen herhangi bir sorgunun en iyi çalışacağı terim ağırlıklandırma modelini önceden belirleyemiyoruz. Terim ağırlıklandırma modellerinin başarımı üzerine etki eden sorgu ve derlem karakteristikleri hakkında çok az bilgiye sahibiz. Bu doktora tezinde, söz konusu gizeme bir nebze olsun ışık tutmak amaçlanmaktadır.

Bu tezde sunulan bütün deney sonuçlarını tekrarlamak ve yeniden üretmek için gerekli olan veri ve kod çevrimiçi olarak mevcuttur.

Anahtar Sözcükler: Ki-Kare Testi, İndeks Terim Ağırlıklandırma, Frekans Dağılımı, Bilgi Erişimde Başarım Gürbüzlüğü Problemi, Seçkili Bilgi Erişim.

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my supervisor, Associate Professor Dr. Bekir Taner Dinçer, who has supported me throughout my dissertation with his patience and unsurpassed knowledge. I thank him for his guidance and encouragement, where good advice, support, and friendship has been invaluable on both academic and personal levels.

Also, I would like to thank Professor Dr. Ümit Deniz Turan, Associate Professor Dr. Cihan Kaleli, Assistant Professor Dr. Kemal Özkan and Assistant Professor Dr. Mehmet Koç on my dissertation committee for their valuable contributions.

Finally, I dedicate this PhD dissertation to `the projects` from which I came...

Ahmet ARSLAN

August, 2016

This work is supported by TÜBİTAK, scientific and technological research projects funding program, under grant 114E558. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

19/08/2016

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Anadolu University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Ahmet ARSLAN

CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

GLOSSARY OF SYMBOLS AND ABBREVIATIONS

$\aleph$	Aleph
χ^2	Chi Square
$\dagger$	Dagger
γ	Gamma
λ	Lambda
μ	Mu
ω	Omega
ρ	Rho
σ	Sigma
AQE	Automatic Query Expansion
BM	Best Match
CDF	Cumulative Distribution Function
CE	Collection Enrichment
DFI	Divergence from Indepence
DFR	Divergence from Randomness
ECIR	European Conference on Information Retrieval
ERR	Expected Reciprocal Rank
GOF	Goodness of Fit
HTML	Hyper Text Markup Language
ICTF	Inverse Collection Term Frequency

IDF	Inverse Document Frequency
IR	Information Retrieval
k-NN	k-Nearest Neighbors
LETOR	Learning to Rank
MAP	Mean Average Precision
MLE	Maximum Likelihood Estimate
MDS	Multidimensional Scaling
MQ	Million Query
MS	Model Selection
NIST	National Institute of Standards and Technology
NDCG	Normalized Discounted Cumulative Gain
PDF	Probability Density Function
PMF	Probability Mass Function
PRP	Probability Ranking Principle
RIA	Reliable Information Access
SE	Standard Error
SIGIR	Special Interest Group on Information Retrieval
SIR	Selective Information Retrieval
STW	Selective Term Weighting
QPP	Query Performance Prediction
TREC	Text REtrieval Conference
URL	Uniform Resource Locator
WT	Web Track

1. INTRODUCTION

Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.

–Samuel Johnson, 1775

1.1. Introduction

With the continuos and rapid growth of the Internet, digital information that is available on the World Wide Web have become enormous. Moreover, the amount of new information being produced is increasing exponentially. It is estimated that a week's worth of the New York Times contains more information than a person was likely to come across in a lifetime in the 18^{th} century (Wurman, 2000).

There were the times when the philosophers had been used to attain proficiency in multiple and diverse subjects; including politics, physics, biology, religion and mathematics. By contrast, today it is not even possible to know all subjects as individuals. In this context, to *know how to find information about a subject* is way more important than knowing a subject as individuals. That makes advanced search tools the only meaningful way to access to the surging volume of available information. Categorical browsing is simply not possible due to the virtually unlimited size. Search engine tools have already become part of daily routine for everyone, not just professionals such as academicians, librarians and lawyers. There are 3.5 billion searches per day on Google, whose mission is to organize the world's information and make it universally accessible and useful (Sullivan, 2015).

We live today in the *information age*, that is access to and the control of information is the defining characteristic of the current era in human civilization. Categorical browsing is abandoned in favor of search for accessing information and that is why search is key to the *information age*.

1.2. Motivations

Although many term-weighting models have been proposed for information retrieval (IR), most of the current approaches tend to systematically use a single term-weighting model to satisfy every information need of users. Such a term weighting model is usually determined by comparing the *average* effectiveness of the existing models over a given set of queries. A term-weighting model may show a good performance on average, but as it can be seen in Figure 1.1, it is an empirical fact that every term-weighting model shows a large variation in performance across queries. This suggests that by using a single term-weighting model, some particular queries can be satisfied with extremely high performance, while the others are poorly performed. The basic premise for selective term-weighting is that there is no *single* term-weighting model that performs the best on *all* queries. Our main motivations for the present PhD dissertation are as follows:

- Term-weighting models are usually systematically applied to all queries.
- Usually a single term-weighting model that is the most effective on the average is preferred and deployed in a search system.
- Arithmetic mean (average) of traditional effectiveness measures are dominated by the better-performing queries.
- Information needs of users are diverse.
- Every term weighting model may be successful on different queries.
- Even if a single term-weighting succeeds good on the average, it performs very poorly for some queries.
- A single term weighting model is not suitable to satisfy all information needs.
- In the context of information retrieval evaluation, it is important to take into account the per-query performance of term weighting models as well as average performance.

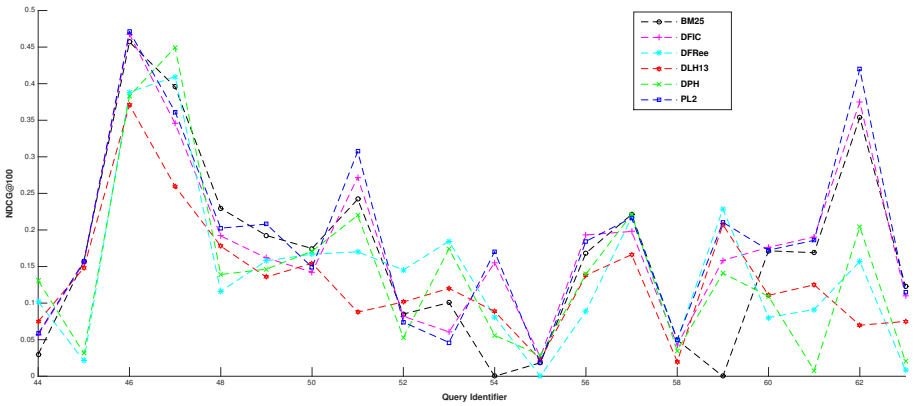

Figure 1.1. The variance in effectiveness (NDCG@100) among eigth models

- Many retrieval systems suffer from high variance in performance across queries: The Problem of Robustness in IR Effectiveness.
- High variance in retrieval effectiveness across queries is undesirable, since users might disappointed by a significant failure of the system.
- End users tend to remember their bad search experiences when interacting with an information search system.
- A disappointed end user may abandon the search system regardless of the system's average performance.
- Therefore, a robust retrieval system, which does not disappoint its users often by minimizing the risk of significant failures, is indeed desirable.

1.3. Research Questions and Hypotheses

The statement of this dissertation is that the most effective term-weighting model can be accurately predicted from a number of candidate term-weighting models for each given query. This is investigated in the context of a framework, called Selective Term Weighting (STW), where the success probability of a term-weighting model for a given query is estimated based on the queries it performed the best and the worst on the already seen test queries. In the selective term weighting framework, the queries that the term-weighting model performed both the best

and the worst are identified from the test query set. The distance of a given test query to the identified query set is computed for each model and model with the highest probability is selected for the given query.

This dissertation analyze frequency distribution of terms on document collections in the context of IR. This work develops a framework to selectively apply an appropriate term weighting model on a per query basis. The prediction of a model among the candidate model set is based on the goodness of fit frequency distribution of query terms. In particular, this dissertation addresses the following research questions:

[R1] For a given query, can the most effective term-weighting model be predicted among the current state-of-the-art models based on the frequency distributions of the queries' terms?

[R2] Can a more effective or robust system be built by per-query application a model predicted among current state-of-the-art technologies than a system in which a single/individual model is uniformly applied to all queries?

Given these research questions, a number of hypotheses can be formally stated and tested.

[H1] For a given query, the most effective term-weighting model can be predicted with reasonable accuracy, by analyzing the frequency distributions of query terms on document collections.

[H2] Different queries benefit differently from each term-weighting model and the retrieval effectiveness and robustness can be significantly enhanced if an appropriate term weighting model is used for each individual query.

1.4. Contributions

The main contributions of this dissertation are the introduction of the STW framework and the proposed use of chi-square goodness-of-fit test on frequency distribu-

tions of query terms for identifying similar queries. In addition, this dissertation draws insights from a large set of experiments, involving three different standard corpora, two different search tasks, two different document representations and two different effectiveness measures calculated at various cutoff levels. This illustrates the generalizability of the STW framework.

Furthermore, we thoroughly evaluate the accuracy, effectiveness and robustness of the STW framework on two different retrieval tracks, namely Web Track and Million Query Track. In particular, a Web collection that contains over a half billion English documents and about one thousand queries are used in this evaluation.

This study makes some important contributions to the body of existing work in both selective IR and robust IR. This dissertation presents experiments of the selective term-weighting for robust retrieval based on frequency distributions of query terms. This is the first examination of the frequency distributions of query terms on document collections in text-based IR. This has not been done before. As a by-product, a new family of query features can be driven from the frequency distribution of query terms for to use in IR research area.

This work presents a unique evaluation methodology for selective retrieval approaches when there exist multiple candidates to choose from. Three aspects of such evaluation: accuracy, effectiveness and robustness are considered at the same time. Two natural baselines that any selective retrieval approach should outperform at the minimum are derived and described.

The present dissertation also reveals the organic connection between the selective IR and the robust IR that focused on avoiding significant failures caused by the poorly-performing queries. This connection has much to do with the true understanding/definition of a significant failure, and an appreciation of it helps to gain insight into the selective retrieval approach.

Indeed, significant failure is a vague concept. When does a retrieval system

fail significantly? Can an effectiveness score of 0.2 or 0.6 be considered a failure for a particular query? Whether a system performs poorly or not can only be meaningfully identified when it is *relatively* compared to the *other* systems. For example, a system serves a query with the effectiveness measures of 0.6 and all the other systems attain effectiveness score greater than 0.7. Since the model in question is the least effective, we can call the score of 0.6 a significant failure. On the other hand, a system can be the most effective with a score of 0.2 when the other systems return zero relevant documents. Obviously effectiveness score of 0.2 is not a significant failure in this case.

These examples clearly demonstrate that significant failure must be defined in a relative manner. There must be other systems to compare with. The magnitude of an effectiveness score alone is not enough to define it. This is where selective retrieval approaches come into play. The interesting relationship between the selective retrieval approaches and the problem of robustness in retrieval effectiveness is that the selective approaches are natural solutions to the robustness problem.

1.5. Outline of the Dissertation

The remainder of this dissertation is organized as follows:

Chapter 2: Information Retrieval This chapter discusses relevant background material in IR. The main stages and models in IR are discussed, as well as the evaluation and the effectiveness measures in general. Term-weighting is shown to be a crucial aspect in IR. Furthermore, the problem of robustness in IR effectiveness is explained.

Chapter 3: Related Work: Selective Information Retrieval This chapter describes the previous selective retrieval approaches proposed in the literature, as well as briefly surveys the existing studies on the query performance prediction.

Chapter 4: Our Approach: Selective Term Weighting This chapter describes the proposed selective term-weighting approach based on frequency distributions of query terms.

Chapter 5: Experimental Methodology This chapter provides the details of the experimental setup.

Chapter 6: Experimental Results and Analysis This chapter presents the experimental results and our interpretations of them.

Chapter 7: Concluding Remarks This chapter discuses concluding remarks and some interesting future directions for further research.

2. INFORMATION RETRIEVAL

2.1. Introduction

Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need (query) from within large collections (usually stored on computers) (Manning et al., 2008). The ultimate goal is to satisfy user's information need. To do so, the IR system returns documents that might contain the desired information. The documents that satisfy user's information needs are called *relevant* documents. An ideal IR system is expected to return only relevant documents.

An IR system typically compromises of four processes: (*i*) indexing, (*ii*) query formulation, (*iii*) matching, and (*iv*) re-ranking. Figure 2.1 shows the process flowchart diagram.

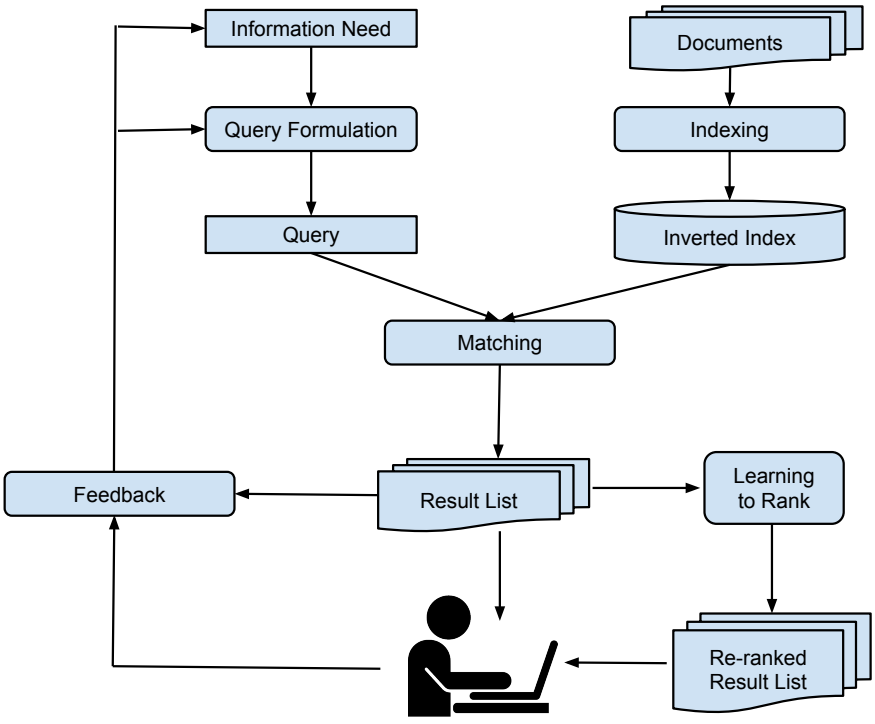

Figure 2.1. Information Retrieval Process

2.2. Indexing

Linear scanning of documents (e.g., '*grep*') would be terribly slow. To speed-up matching process, an off-line process is necessary where documents are saved into an inverted index. Common stages employed by IR systems to derive the index representations are described in the following subsections.

2.2.1. Tokenization

This is where free form text is break into words or tokens. For some systems, as simple as splitting on white spaces will suffice for the task. However, some other systems may need more sophisticated tokenizers (e.g., recognizes e-mail addresses and keeps them as one token). Yet, it could be more troublesome for languages that do not use white space for word boundaries (e.g., Chinese, Japanese, Thai). These languages have to employ *word segmentation* for the tokenization task.

2.2.2. Stop words removal

Some of the words do not constitute in the meaning, but used for grammatical necessities. These extremely frequent words (e.g., "*the*," "for," "of") called *function words* or *stop words*. Not indexing these words is called stop word removal. This will reduce the index size, but it has some drawbacks. For instance, it would not be possible to retrieve any documents for the query "*to be or not to be*." And returned documents would make no sense for certain two-term queries such as "*the current*," "*the wall*," "*the who*," and "*the sun*."

2.2.3. Stemming

Stemming removes (inflectional) suffixes to reduce words to a common base form. For instance, the words "*addicted*," "*addicting*," "*addiction*," "*addictions*," "*addictive*," and "*addicts*" can be reduced to their stem: **addict**. Many stemming algorithms are proposed for to use in IR, probably the most commonly used one is the Porter stemmer (Porter, 1997). KStemming (Krovetz, 1993) is another widely used stemmer for English, which is a less aggressive alternative to the Porter stemmer.

2.3. Query Formulation

There is a distinction between an information need and actual query submitted to the IR system. For example the information need "What folk remedies are there for soothing a sore throat?" can be formulated into the query "folk remedies sore throat." Some search systems guide their users in their query formulations by means of providing feedback; e.g., "*related searches*" or "*did you mean?*" features of commercial search engines.

2.4. Matching

The document representation and the query are compared to produce a result list, in which documents are ranked in decreasing order of relevance. The relevance of a document representation to a given query can be estimated by various IR models. The following subsections will describe models IR.

2.4.1. The Boolean model

The Boolean model is one of the oldest IR models. The model employs the operators of George Boole's mathematical logic (AND, OR, NOT) to combine query terms.

This model lacks ranking mechanism. Documents are either retrieved or not, but the documents in the result set are not ranked. Therefore, all document are assumed equally important.

2.4.2. The vector space model

Salton and McGill (1986) considered the document representations and the query as vectors defined in a high dimensional Euclidean space. Each term represented by a separate dimension, thus dimension of the space is equals to total number of unique terms (denoted by N) in the index. Equation 2.1 is the cosine of the angle θ between the two vectors $\overrightarrow{d}$ and $\overrightarrow{q}$, which is used to estimate relevance.

$$score(\vec{d}, \vec{q}) = \cos(\theta) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}| \times |\vec{q}|} = \frac{\sum_{i=1}^{N} d_i \times q_i}{\sqrt{\sum_{i=1}^{N}(d_i)^2} \times \sqrt{\sum_{i=1}^{N}(q_i)^2}} \tag{2.1}$$

It should be noted that $\cos(0°) = 1$ and $\cos(90°) = 0$. In the vector space model, the values of the vector components are not defined. The problem of assigning appropriate weights to the vector components is known as **term weighting**. Probably the most famous term weighting is the $tf \cdot idf$ weights, which is a combination of within-document term frequency tf and inverse document frequency idf.

$$weight(t, D) = tf(t, D) \times \log_2 \frac{N}{df(t)} \tag{2.2}$$

Many modern weighting algorithms are based on the concepts in $tf \cdot idf$ weighting.

2.4.3. The 2-Possion model and best match weighting

The probabilistic retrieval model ranks the documents in the collections in order of decreasing probability of relevance $P(R|D)$, that is the probability of relevance R given the document D. Robertson (1997) turned the idea of ranking by the probability of relevance into the Probability Ranking Principle (PRP).

The general form of the PRP is given in Equation 2.3, which was proposed by Robertson and Jones (1976) and named as the RSJ model.

$$S(Q, D) = \log P(R|D) = \sum_{t \in Q \cap D} \frac{P(D_t = 1|R) \cdot P(D_t = 0|\bar{R})}{p(D_t = 0|R) \cdot P(D_t = 1|\bar{R})} \tag{2.3}$$

Here, R is a random variable who takes the values $\{R, \bar{R}\}$, where $R =$ relevant and $\bar{R} =$ non-relevant. $P(R)$ denotes probability of relevance, while $P(\bar{R})$ denotes probability of non-relevance. D_t is another random variable who takes the values $\{0, 1\}$, where $D_t = 0$ means the document D does not contain the term t and $D_t = 1$ means the document D contains the term t. So, $P(D_t = 1|R)$ is the probability

that the document D contains the term t given relevance. Robertson and Walker (1994) assumed a 2-Poisson model for these distributions and developed the famous Okapi BM25 term-weighting model, which is still one of the best performing term-weighting algorithms.

The BM25 model scores a document-query pair using the following formula:

$$score(D,Q) = \sum_{t \epsilon Q \cap D} IDF(t).\frac{tf_{t,D} \cdot (k_1 + 1)}{tf_{t,D} + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})} \tag{2.4}$$

where k_1 and b are free parameters that control the term frequency saturation and the document length normalization respectively. Since BM25 contains two free parameters (k_1 and b) in fact it represents a number (or family) of term-weighting schemes in each case. A specific term-weighting scheme is only recovered by setting the free parameters to specific values.

2.4.4. Language models

Language models (LM) have been successfully used in the automatic speech recognition systems. In particular, the LM is used to choose the most probable text from the candidate texts generated by the acoustic model. For example, candidate texts would be "male infertility" and "mail infertility" and then, the LM would choose the correct one, which is "male infertility." Because "male infertility" has much higher probability to occur in the English language.

Application of language models to IR (Ponte and Croft, 1998; Hiemstra, 2000; Lafferty and Zhai, 2001) is a more recent innovation, which is actually borrowed from the automatic speech recognition domain. In LM approach to IR, every document has its own LM. Then the probability of the query being generated by that document LM is calculated for each document. In case of a *unigram* model, this is the multiplication of probabilities of individual terms, as given by Equation 2.5. Then this probability is used to rank documents for a given query.

$$P(t \in Q|D) = \prod_{t \in Q} P(t_i|D) \quad where \quad P(t|D) = \frac{tf_{t,D}}{|D|} \tag{2.5}$$

However, Equation 2.5 assigns zero probability to a document that does not contain all of the query terms ($t \in Q$). To avoid a zero probability, usually a method called *smoothing* is employed, in which some non-zero probability is assigned to any term that does not occur in the document being scored. Zhai and Lafferty (2004) studied smoothing methods for LM applied to IR, such as Jelinek-Mercer, Dirichlet prior and Absolute discount. The Dirichlet prior for smoothing is known to be the most effective (Croft et al., 2009).

2.4.5. Divergence from randomness

Amati and Van Rijsbergen (2002) introduced the Divergence From Randomness (DFR) framework, where it is assumed that the important terms of a document are the terms whose frequencies diverge from the frequency suggested by a basic randomness model, such as Poisson, Hyper-Geometric, Bose-Einstein etc. The probabilistic modular framework deploys more than 50 term-weighting models. One of the most popular DFR models is PL2, which is particularly effective at high precision tasks. PL2 assumes a Poisson distribution for the normalized term frequency (tfn) distributions and employs Normalization2 (given in Equation 2.6) to within-document term frequency ($tf_{t,D}$).

$$tfn = tf_{t,D} \cdot \log_2(1 + c \cdot \frac{avdl}{|D|}) \tag{2.6}$$

where c is a free parameter and $avdl$ is the average document length in the collection. The PL2 model scores a document-query pair using the following formula:

$$PL2(D,Q) = \sum_{t \in Q \cap D} \frac{1}{tfn+1}\Big(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn) \cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn)\Big) \tag{2.7}$$

where λ is the variance and mean of a Poisson distribution. λ is given by within-collection term frequency divided by the total number of documents in the collection.

2.4.6. Information-based models

Clinchant and Gaussier (2009, 2010, 2011) introduced the family of information-based models for ad hoc IR. These models draw their inspiration from a long-standing hypothesis in IR, namely the fact that the difference in the behaviors of a word at the document and collection levels brings information on the significance of the word for the document.

The most successful instantiation of the model is called LGD, which assumes a log-logistic distribution for the normalized term frequency (tfn) distributions.

$$LGD(D,Q) = \sum_{t \in Q \cap D} -\log\left(\frac{\lambda_t}{\lambda_t + tfn}\right) \quad where \quad tfn = tf_{t,D} \cdot \log_2\left(1 + c \cdot \frac{avdl}{|D|}\right) \quad (2.8)$$

Note that LGD employs the same term frequency normalization (Normalization2 given in Equation 2.6) as PL2. Thus LGD and PL2 share the same free-parameter c in common.

2.4.7. Divergence from independence

Kocabaş, Dinçer and Karaoğlan (2014) introduced an out-of-the-box automatic term weighting method which is based on measuring the degree of divergence from independence (DFI) of terms from documents in terms of their frequency of occurrence. DFI is the $non-parametric$ counterpart of DFR and it has a well-establish underling statistical theory.

DFI calculates an *expected* term frequency of a term t in a given document D. The document collection is considered as one big document C whose document length $|C|$ is the total number of terms in the entire collection. The term t is

observed tf(t,C) times in the artificial document whose length is $|C|$. Using rates and ratios, an expected term frequency e is calculated by the Equation 2.9.

$$\frac{tf(t,C)}{|C|} = \frac{e}{|D|} \tag{2.9}$$

where tf(t,C) is the within-collection term frequency of the term t. Then the expected term frequency e is compared with the actual observed within-document term frequency tf(t,D). If $tf(\mathrm{t,D}) \leq e$ then DFI returns zero score. Elsewhere there are three basic measures of DFI, each of which arises from different bases:

- DFIB $= \log_2\left(\frac{tf(t,D)-e}{e}+1\right)$ based on *saturated model of independence*
- DFIC $= \log_2\left(\frac{(tf(t,D)-e)^2}{e}+1\right)$ based on *normalized chi-squared distance* from independence
- DFIZ $= \log_2\left(\frac{tf(t,D)-e}{\sqrt{e}}+1\right)$ based on *standardization*

In brief, the DFIZ is good at tasks that require high recall whereas DFIC is good at tasks that require high precision.

2.5. Re-ranking

Re-ranking is a process that is employed after the matching process. In matching process, a standard term-weighting model (BM25, PL2, LGD, etc) returns a list of documents for a given query. Re-ranking the top-K documents (*sample*) in the list returned by the *reference* term-weighting model using machine learning techniques is called Learning to Rank (Liu, 2009). Learning to rank involves the deployment of various features extracted for a sample of documents into effective learned models, which is then used to re-rank (Macdonald et al., 2013b). Quite a number of features combined within an effective learned model: including both query independent document features (incoming links, URL depth, etc) and query dependent features which are the weights/scores assigned to fields of documents

Table 2.1. IR Evaluation Forums

Forum	Name	Reference
TREC	Text Retrieval Conference	`trec.nist.gov`
CLEF	Conference and Labs of the Evaluation Forum	`www.clef-initiative.eu`
FIRE	Forum for IR Evaluation	`fire.irsi.res.in`
NTCIR	NII Testbeds and Community for for Information Access Research	`ntcir.nii.ac.jp`
INEX	Initiative for the Evaluation of XML Retrieval	INEX has come to an end
ROMIP	Russian IR Evaluation Seminar	`www.romip.ru`

(title, body, anchor text, etc) by multiple term-weighting models (BM25, $tf \cdot idf$, etc) for query terms (Macdonald et al., 2013a).

Learning to rank has been gaining considerable attention in the IR community, given there is an increasing amount of research has been devoted to develop and compare learning to rank methods (Tax et al., 2015).

2.6. Evaluation in Information Retrieval

The evaluation of IR systems is the process of assessing how well a system satisfies the information needs of its users (Voorhees, 2002). IR Evaluation Forums create test collections and provide standardized evaluation of IR systems. The major six forums are listed in Table 2.1. Probably the most famous one is the Text REtrieval Conference (TREC), which is co-sponsored by the National Institute of Standards and Technology (NIST) and United States Department of Defense.

To comparatively evaluate IR systems, primarily, the traditional TREC-style (also referred to as Cranfield paradigm) evaluation methodology is adopted (Voorhees and Harman, 2005). The evaluation methodology requires a document collection, a set of information needs (called topics or queries), and a set of relevance judgments (right answers) indicating which documents are relevant to which topics (Voorhees, 2007). An example of an information need (topic) and excerpt from query relevance judgments of the example topic are given in Listing 2.1 and Listing 2.2 respectively.

Listing 2.1. Example of an Information Need

```
<topic number="300" type="single">
  <query>how to find the mean</query>
  <description>
   Find a page that explains how to compute the mean of a set of numbers.
  </description>
</topic>
```

Listing 2.2. Excerpt from Query Relevance Judgments

```
300 0 clueweb12-1810wb-14-05198 0
300 0 clueweb12-1811wb-49-13206 2
300 0 clueweb12-1811wb-95-05256 1
300 0 clueweb12-1812wb-00-27455 1
```

The query relevance judgments (`qrels`[1]) file consists of four columns: TOPIC#, ITERATION, DOCUMENT# and RELEVANCY.

1. TOPIC# is the topic number
2. ITERATION is the feedback iteration (almost always zero and not used)
3. DOCUMENT# is the official document identifier
4. RELEVANCY is an integer code where less than one indicates non-relevant and greater than zero indicates relevant.

Relevance labels, which can be either binary (1=relevant or 0=non-relevant) or graded (2=highly relevant; 1=relevant; 0=non-relevant; -2=spam/junk), are assigned to each query-document pair by the human assessors therefore it is a time and labor expensive task. For this reason, TREC employs a process called *pooling* in which the human assessors judge only the documents that are the union of the set of top-k (usually 100) retrieved documents for each topic by the TREC participants.

[1] http://trec.nist.gov/data/qrels_eng

TREC participants return a ranking of the documents in the collection in order of decreasing probability of relevance for each information need. The top-N (usually 1000) documents for each query in the topic set are saved into a submission file, which produce an experimental run. An excerpt from a submission file, which consists of six columns per line, is given in Listing 2.3.

Listing 2.3. Example of TREC submission file format

```
300 Q0 clueweb12-1712wb-85-10084 1 4.704471 BM25k1.8b0.2_KStem
300 Q0 clueweb12-0102wb-39-28701 2 4.627909 BM25k1.8b0.2_KStem
300 Q0 clueweb12-0307wb-58-22743 3 4.454082 BM25k1.8b0.2_KStem
300 Q0 clueweb12-0109wb-82-21760 4 4.447290 BM25k1.8b0.2_KStem
300 Q0 clueweb12-0307wb-28-01469 5 4.389167 BM25k1.8b0.2_KStem
```

1. column is the topic number.
2. column is unused and should always be "Q0."
3. column is the official document identifier of the retrieved document.
4. column is the rank of the document that is retrieved.
5. column is the relevancy score of the document.
6. column is called the "run tag" that corresponds to a unique identifier for the retrieval method used.

2.6.1. Evaluation measures of retrieval effectiveness

To quantify retrieval effectiveness, several evaluation measures have been proposed. *Precision* and *recall* were the two simple *set-based* measures developed early on. Precision is the fraction of retrieved documents that are relevant while recall is the fraction of relevant documents that are retrieved.

$$precision = \frac{\#\ retrieved\ documents\ that\ are\ relevant}{\#\ retrieved\ documents} \tag{2.10}$$

Table 2.2. Contingency Table

	Relevant	**Non-relevant**
Retrieved	true positive	false positive
Not retrieved	false negative	true negative

$$recall = \frac{\#\ retrieved\ documents\ that\ are\ relevant}{\#\ relevant\ documents\ in\ the\ entire\ collection} \tag{2.11}$$

Precision and recall can also be defined/expressed in terms of true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn); which are borrowed from the classification domain. Table 2.2 is called the *contingency* table whose cells are the four possible combinations. Precision and recall definition in terms of combinations of the contingency table's cells are as follows:

$$precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn} \tag{2.12}$$

Precision decreases as the number of retrieved documents increases. Recall increases as the number of retrieved documents increases. Note that, recall computation requires the total number of relevant documents in the entire collection, which is impossible to know for very large data bases.

The F-measure combines scores for precision and recall by employing the *harmonic* mean into a single measure to allow the comparison of IR systems.

$$F_{measure} = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.13}$$

Precision and recall are set-based measures because they do not consider ordering of documents that are retrieved. For example consider two result lists $R = \{0, 0, 1, 1, 1\}$ and $S = \{1, 1, 1, 0, 0\}$ where 0 indicates a non-relevant document, 1 indicates a relevant document. Since they both return five documents, three of which are relevant, their precision will be the same value of $\frac{3}{5}$. In other words, set-based measures cannot distinguish/differentiate R and S. By contrast, following

Table 2.3. Six Shades of Relevance

Grade	4	3	2	1	0	-2
Abbr.	Nav	Key	HRel	Rel	Non	Junk

rank-based measures would favor the result list S that returns relevant documents higher in the ranked list.

Mean Average Precision (MAP): Precision at a fixed rank k (P@k) is the fraction of relevant documents among top-k results, for example Precision at rank 20 (P@20). Average precision (AP) is the average of the precision scores calculated after each relevant document is retrieved, using zero as the precision for relevant documents that are not retrieved (Buckley and Voorhees, 2000).

$$AP(q) = \frac{1}{|R_q|} \sum_{r \in R_q} P@rank(r), \tag{2.14}$$

where R_q represents the documents that are relevant to the query q. For the previous example, AP of S will be $AP_S = \left(\frac{1}{1} + \frac{2}{2} + \frac{3}{3}\right) \div 3$, which is greater than the AP value of R will be $AP_R = \left(\frac{1}{3} + \frac{2}{4} + \frac{3}{5}\right) \div 3$.

AP(q) is usually computed using a raking truncated at 1000. When AP is averaged over the whole query set ($q \in Q$), MAP is obtained. MAP is a standard metric for binary relevance assessments. More recently, six-point grading scale has been used to judge document-query pairs at NIST. Detailed descriptions of six different relevance grades, which presented on Table 2.3, can be found in the TREC 2014 Web Track overview report (Collins-Thompson et al., 2015). Unlike the binary effectiveness measure MAP, which treats relevance grades 1/2/3/4 as relevant and grades 0/-2 as non-relevant, following two retrieval effectiveness measures are based on graded relevance judgments.

Normalized Discounted Cumulative Gain (NDCG@k) : It is a widely used measure that can handle graded relevance judgments as defined by Järvelin and

Kekäläinen (2002). DCG at rank k is computed as:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{g_i} - 1}{log_2(1+i)}, \tag{2.15}$$

where g_i is the relevance grade of the document at rank i. NDCG is the normalized version of Eq. 2.15 and is calculated as $\frac{DCG@k}{ideal\,DCG@k}$.

Expected Reciprocal Rank (ERR@k): It discounts the documents that are shown below very relevant documents, and is defined as the expected reciprocal length of time that a user will spend to find a relevant document. The computation of ERR@k as defined by Chapelle, Metlzer, Zhang and Grinspan (2009) is follows:

$$ERR@k = \sum_{i=1}^{k} \frac{R(g_i)}{i} \prod_{j=1}^{i-1} (1 - R(g_j)), \tag{2.16}$$

where $R(g) = \frac{2^g - 1}{16}$ and $g_1, g_2, \ldots, g_k$ are the relevance grades associated with the top k documents.

MAP and NDCG reflect the overall performance (top-k documents) of the systems, while ERR is precision biased. ERR@20 and NDCG@20 can leverage graded relevance judgment presented on Table 2.3 (except that a value of -2 is treated as 0) and are the two standard retrieval effectiveness metrics used at recent TREC Web Tracks (Collins-Thompson et al., 2015).

2.6.2. Evaluation scripts for measure calculation

Another important component supplied by the TREC organizers is the standard evaluation tools to calculate the effectiveness of a retrieval run. This is important for setting a standard in retrieval effectiveness measure calculation. Without being extensive, the most mainstream evaluation tools are as follows:

`trec_eval`[2] is the very first evaluation script published by the TREC organizers for evaluating an ad hoc retrieval run, given the submission/result file

[2] `http://trec.nist.gov/trec_eval/`

and a standard set of judged results (`qrels`). The tool reports a wide range of effectiveness measures over the run.

`gdeval.pl` is the evaluation tool for calculating NDCG@k and ERR@k measures, which can be downloaded from trec-web-2014[3] GitHub repository.

`statAP_MQ_eval_v4.pl`[4] is the evaluation script used in the Million Query (MQ)[5] tracks of TREC, in which query relevance judgments are published as five-column `prels` file format instead of the traditional four-column `qrels` file format. The five columns of the `prels` file format are as follows: TOPIC#, DOCUMENT#, ALGORITHM# and INCLUSION_PROBABILITY. The script computes NDCG@{10,30,50,100} and Statistical Average Precision (statAP) as defined by Carterette, Pavlu, Kanoulas, Aslam and Allan (2008).

Many researchers have been using these tools and viewed them as an official/reliable implementation of the various retrieval effectiveness measures proposed in the literature.

2.6.3. Summing up

The described test collection-based evaluation and experimentation methodology (Sanderson, 2010) permit researchers to easily compare IR systems based on their retrieval effectiveness. Fang et al. (2004, 2011) have proposed an alternative evaluation methodology to *analytically* and *experimentally* diagnose the weaknesses or strengths of IR models. Their major contribution is the retrieval heuristic constraints, which are defined independently of relevance judgments, so that we can study and compare IR models analytically without requiring experimentation.

The present dissertation follow the TREC evaluation standards, as many researches do. Standard benchmark datasets and query sets are keys for establishing research to be reliable, reproducible and extensible for the future. But there are known limitations of the TREC-style evaluation, which are out of the scope of the

[3] `http://github.com/trec-web/trec-web-2014`
[4] `http://ir.cis.udel.edu/million/statAP_MQ_eval_v4.pl`
[5] `http://trec.nist.gov/data/million.query.html`

present dissertation.

- Query set is too few to make generalizable inferences.
- Due to the pooling concept, reusability of the datasets and query relevance judgements are questioned (Buckley et al., 2006).
- Does the sample query set really represent the population?
- Does the document set really represent the Web?

2.7. The Problem of Robustness in Retrieval Effectiveness

The robustness problem of IR is caused by a large/radical fluctuation of/in a system's retrieval effectiveness across ad hoc queries posed by the users, as measured by the quality of returned documents. Even if a retrieval system succeeds very well on average, to the quality of returned documents for certain queries is poor (significant failure of the system). These poorly-performing queries may lead to user dissatisfaction since people tend to remember their *bad* experiences. In other words, the system's average performance does not help a user who is disappointed by a significant failure of the system. Thus, a robust system that avoids significant failures, as well as performs very well on the average, is desirable.

The robustness problem has been recognized by the IR community (Carmel and Yom-Tov, 2010, Chapter 1) and led to a new research direction that focuses on the poorly-performing queries and examines/explores new evaluation measures that are not dominated by the better-performing topics. We will give brief information about a workshop and TREC tracks dedicated on the robustness.

2.7.1. The reliable information access workshop

The Reliable Information Access (RIA)[6] Workshop was held in the summer of 2003 (Harman and Buckley, 2004). It was the first attempt to rigorously analyze

[6] `https://ir.nist.gov/ria`

individual query failures and understand the reasons for performance variability between queries and systems. The goal of the RIA workshop was to learn how to customize IR systems for optimal performance on any queries. The workshop brought together seven different IR systems and assigned them to common IR tasks. By performing extensive topic failure analysis, nine failure categories were identified.

A surprising result was the finding that the majority of failures could be fixed with traditional IR techniques such as better relevance feedback mechanism and better query analysis. Harman and Buckley (2009) argued that "it may be more important for research to discover what current techniques should be applied to which topics, rather than to come up with new techniques."

It is observed that some retrieval approaches work well on one topic but poorly on a second, while other approaches may work poorly on the first topic, but succeed on the second. Buckley (2009) stated that: "if one could determine in advance which approach would work well, then a dual approach could strongly improve performance. Unfortunately, no one knows how to choose good approaches on a per-query basis."

The findings and conclusions drawn from the RIA workshop by Harman and Buckley (2004, 2009), have driven a new research direction in the IR field on selectively applying retrieval approaches on a per-query basis. This research area is later on termed as *selective* information retrieval.

2.7.2. The TREC 2003-2005 robust retrieval track

The fluctuation in retrieval effectiveness across queries and systems led to the TREC robust retrieval tracks in the years 2003-2005, which explored methods for improving the consistency of retrieval technology by focusing on poorly performing topics (Voorhees, 2005). Systems were challenged by 50 old TREC topics found to be *difficult* for most systems over the years. A topic is considered difficult in this context when the median of the AP scores of all participants for that topic is below

a given threshold (i.e., half of the systems are scored lower than the threshold), but there exists at least one high outlier score.

Since traditional measures are dominated by the better-performing topics, the track has also investigated appropriate evaluation measures that emphasize a system's least effective topics. A variant of the traditional MAP measure that uses a geometric mean (which gives appropriate emphasis to poorly performing topics) to average individual topic results is developed during the lines of robust retrieval tracks. Participants tried approaches to decrease the variance in retrieval effectiveness across the topic set by increasing retrieval effectiveness for poorly-performing topics. The most promising approach is found to be exploiting external collections other than the target collection such as the Web.

2.7.3. The TREC 2013-2014 risk-sensitive retrieval task

In 2013, the TREC forum introduced a new risk-sensitive retrieval task (Collins-Thompson et al., 2014) which rewards algorithms that achieve improvements in terms of their average effectiveness across topics (good performance on average), but that also maintain good robustness. Robustness of a system is defined as *minimizing the risk of significant failure* relative to a given baseline but also achieving good average effectiveness over all topics at the same time.

The goal of the risk-sensitive task is two-fold:

1. "to encourage research on algorithms that go beyond just optimizing average effectiveness in order to effectively optimize both effectiveness and robustness, and achieve effective tradeoffs between these two competing goals"

2. "to explore effective risk-aware evaluation criteria for such systems"

The risk-sensitive retrieval task is motivated by the empirical fact that the retrieval strategies usually improve the effectiveness for specific queries while degrading it for others compared with a baseline system that does not use such strategies.

Two risk-aware evaluation criteria: *URisk* (Wang et al., 2012) and *TRisk* (Dinçer et al., 2014) were evolved for the task.

The risk-sensitive retrieval task is closely related to the goals of the earlier robust retrieval tracks, thus it can be thought of as a next step in obtaining good retrieval robustness by means of learning to rank techniques.

2.8. Summary

In this chapter, we have presented an overview of IR in general, from indexing to matching documents and evaluation, and reviewed several models of IR such as BM25, DFI, LGD, Language Modeling, $tf \cdot idf$ as well DFR framework. Finally, we call attention to the problem of robustness in IR effectiveness and list TREC tracks and a workshop dedicated on it.

3. RELATED WORK

3.1. Introduction

The previous chapter presented the general background and preliminaries on Information Retrieval (IR). In this chapter, we review the two lines of research that provide the necessary context for the present dissertation: (*i*) Query Performance Prediction (QPP) and (*ii*) Selective Information Retrieval (SIR). Although the present dissertation is not directly related to QPP, we include a brief review of QPP because certain selective retrieval approaches are based on it.

3.2. Query Performance Prediction

Estimating the effectiveness of a search performed in response to a query in the *absence of relevance judgments* is the goal of query-performance prediction methods. Estimating the query difficulty is an important field of study since it enables IR systems to identify *difficult* queries in order to handle them properly. Thus, search engines will reduce the variance in performance, resulting in better retrieval robustness. In this section, we describe the query performance prediction (also referred to as query difficulty estimation) problem and the most successful approaches for this problem in the literature. Following QPP studies' summarizations are based heavily on the book written by Carmel and Yom-Tov (2010). QPP methods can be studied in two categories : pre-retrieval and post-retrieval.

3.2.1. Pre-retrieval predictors

Pre-retrieval predictors estimate the performance of a query before the retrieval takes place, thus, independent of the result list. By contrast, they are collection dependent and analyze the distribution of the query term frequencies within the collection. The inverse document frequency (IDF) and the inverse collection term frequency (ICTF) are frequently used term statistics. The IDF_{avg} and the ICTF_{avg} predictors measure the average of the IDF and ICTF values of the query terms.

The assumption is that queries with high average value, i.e., queries composed of infrequent terms, are easier to satisfy.

He and Ounis (2004a) study a set of predictors of query performance, which can be generated prior to the retrieval process. The linear and non-parametric correlations of the predictors with query performance are thoroughly assessed on the TREC disk4 and disk5 (minus CR) collections. Their research revealed that some of the proposed predictors have significant correlation with query performance, showing that these predictors can be useful to infer query performance in practical applications.

Zhao, Scholer and Tsegay (2008) propose a new family of pre-retrieval predictors based on information at both the collection and document level. The *collection query similarity* predictor measures the vector-space based query similarity to the collection, while considering the collection as a one large document composed of concatenation of all the documents. The *VAR(t)* predictor measures the variance of the term weights over the documents containing it in the collection. The weight of a term that occurs in a document is determined by the specific term-weighting model. If the variance of the term weight distribution is low, then the retrieval system will be less able to differentiate between highly relevant and less relevant documents, and the query is tend to be more difficult.

Hauff, Hiemstra and de Jong (2008), in their survey, categorize and assess 22 pre-retrieval predictors on three different TREC test collections.

3.2.2. Post-retrieval predictors

Post-retrieval predictors require the computation of result list and relevance scores for the query, which is time-consuming. However, these methods are more suitable for identifying inconsistency, incoherency, and other characteristics that reflect low quality.

The pioneering work of Cronen-Townsend, Zhou and Croft (2002) gave rise to query difficulty estimation research. They develop a method for predicting

query performance by computing the relative entropy between a query language model and the corresponding collection language model. The resulting *clarity score* measure was the very first query performance predictor. After 10 years, Hummel, Shtok, Raiber, Kurland and Carmel (2012) presented novel interpretation of *clarity score* and showed that it actually quantifies diversity property of the result list. Their study, along with empirical evaluation, explained the low prediction quality of clarity score for large-scale Web collections.

Yom-Tov, Fine, Carmel and Darlow (2005) won SIGIR'05 best paper award with their "Learning to estimate query difficulty" titled work. They tried to identify *difficult* queries that return poor results, and list several useful use-case scenarios for detection. Estimation is based on the agreement between the top results of the full query and the top results of its sub-queries.

Zhou and Croft (2007) proposed *Query Feedback* for measuring the robustness of the result list to small modifications of the query. If small changes to the query result in large changes to the search results, then the query is considered difficult.

The *Weighted Information Gain* (Zhou and Croft, 2007) measures the divergence between the mean retrieval score of top-ranked documents and that of the entire corpus. The *Normalized Query Commitment* (Shtok et al., 2012) measures the normalized standard deviation of the top scores.

3.3. Selective Information Retrieval

Classical/traditional (non-selective) IR approaches apply a particular technique uniformly to all queries. In contrast, selective retrieval approaches deal with applying different retrieval techniques for different queries. Various selective retrieval approaches have previously been proposed in the literature. In this section, comprehensive survey of existing selective retrieval approaches is given in chronological order. Following SIR studies' summarisations are based heavily on the contents (abstract, introduction, and conclusion) of the cited original works.

3.3.1. Query type classification

Query type classification classifies a query into one of a set of target types (e.g. informational, navigational, or transactional), and then selectively applies a retrieval model trained for the predicted type. For instance, Kang and Kim (2003) showed that different query types can benefit from the application of different retrieval approaches.

3.3.2. Selective weighting function

He and Ounis (2003) tested selective weighting function approach to improving the effectiveness of poorly-performing quires at Robust Track of TREC. He and Ounis (2003, 2004b) were the first to selectively apply a term-weighting model on a per-query basis and they referred to the problem/task as the ***model selection***.

The DFR framework offers over the 50 different term-weighting models, but the framework does not have a strategy to single out one that would yield the best retrieval effectiveness for a given query. He and Ounis (2003, 2004b) proposed a query-based pre-retrieval approach that automatically selects the best-performing retrieval model among 11 DFR models. They cluster the queries according to their statistics and associate the best-performing term-weighting model to each cluster. Their selective approach, which is detailed on Chapter 5.4.2, does improve the poorly-performing queries compared to a baseline where a unique retrieval model is applied indifferently to all queries.

3.3.3. Selective query expansion

Automatic query expansion (AQE) works only for easy queries, i.e., when the search engine is able to rank high the relevant documents. If this is not the case, AQE will add irrelevant terms, causing a decrease in performance. Thus, it is not beneficial to use AQE for every query. Instead, it is advantageous to have a switch that will estimate when AQE will improve retrieval, and when it would be detrimental to it. Amati, Carpineto and Romano (2004) set a threshold on the predicted difficulty,

beyond which queries would be expanded. In this approach, only "easy" queries, i.e., those with highly predicted performance, are expanded. In contrast, a classifier was trained in (Yom-Tov et al., 2005) to identify queries for which pseudo relevance feedback might be beneficial, based on a training set where queries were assessed as to the increase or decrease in performance caused by expansion.

3.3.4. Selective document representation

Plachouras et al. (2004, 2006) investigated the effectiveness of a decision mechanism for the selective combination of evidence in the context of topic distillation task, which is defined as finding useful entry points to sites that are relevant to the query topics. They used three different sources of evidence: textual content of documents, anchor text, and the length of the URL. They concluded that, the selective combination of evidence on a per-query basis can increase the retrieval effectiveness, compared to the uniform combination of evidence (irrespectively of the queries) for Web IR, and more specifically, for topic distillation.

3.3.5. Selective Web information retrieval

Plachouras (2006), in his PhD thesis, proposed a method to selectively apply an appropriate retrieval approach for a given query, which is based on a Bayesian decision mechanism. Features such as the link patterns in the retrieved document set and the occurrence of query terms in the documents were used to determine the applicability of the retrieval approaches. This method was shown to be effective when there were only two candidate retrieval approaches. However, the retrieval performance obtained using this method only improved slightly and actually decreased when more than two candidate retrieval approaches were used.

3.3.6. Selective personalization

Personalization only improves the results for some queries, and can actually harm other queries. Teevan, Dumais and Liebling (2008) characterized queries by using a variety of features of the query, the results returned for the query, and people's interaction history with the query. Using these features they learned Bayesian

dependency networks to identify queries that can benefit from personalization.

3.3.7. Selective search engine

Any given Web search engine may provide higher quality results than others for certain queries. Therefore, it is in users' best interest to utilize multiple search engines. White, Richardson, Bilenko and Heath (2008) propose and evaluate a framework that maximizes users' search effectiveness by directing them to the engine that yields the best results for the current query. Different from previous work on meta-search, they facilitate simultaneous use of individual engines. They describe a machine learning approach (maximum-margin averaged perceptron) to supporting switching between search engines (Google, Yahoo!, and Live Search) and demonstrate its viability at tolerable interruption levels.

3.3.8. Query dependent ranking

Geng, Liu, Qin, Arnold, Li and Shum (2008) proposed a query-dependent ranking approach. They use soft classification that identifies similar queries from a training set. This is different than hard classification which classifies a query into a predefined target class. In their approach, a k-nearest neighbor classifier was used to identify training queries similar to an unseen query. A retrieval model was then learnt based on the identified queries and applied to the unseen query.

3.3.9. Selective query-independent features

Peng and Ounis (2009) investigate a novel approach that applies the most appropriate query-independent feature on a per-query basis. The approach is based on an estimate of the divergence between the retrieved document scores' distributions prior to, and after the integration of a query-independent feature. Experimental results demonstrate that the selective application of a query-independent feature on a per-query basis is very effective and robust.

3.3.10. Selective collection enrichment

Peng, He and Ounis (2009a) proposed a decision mechanism to decide whether or not to apply collection enrichment (CE) on a per query basis. A query performance predictor was used in decision. The approach is based on the predicted performance score of a given query on the local and external resources. In particular, the decision mechanism applies collection enrichment if and only if the predicted query performance score obtained on the external resource is higher than a threshold, as well as the predicted query performance score obtained using the local resource.

Peng, Macdonald, He and Ounis (2009b) we apply the divergence-based approach for selectively applying CE by examining the divergence between relevance score distributions prior to, and after the application of CE. To achieve this, they learn the distribution of divergence scores, which are estimated between two different lists of ranked documents obtained with and without the application of CE, using training data.

3.3.11. Selective diversification

Santos, Macdonald and Ounis (2010) use a large pool of query features to choose between a more lenient or more aggressive diversification strategy on a per query basis. Thorough experiments using the TREC ClueWeb09 collection show that proposed selective approach can significantly outperform a uniform diversification for both classical and state-of-the-art diversification approaches.

3.3.12. Selective ranking function

Peng (2010), in his PhD thesis, proposed the "Learning to Select" framework that selectively applies an appropriate ranking function on a per-query basis, regardless of the given query's type and the number of candidate ranking functions.

Peng, Macdonald and Ounis (2010) choose a ranking function from a large pool of candidate functions, based on their performance on neighboring training queries to an unseen query. The approach employs a query feature to identify

similar training queries for an unseen query. A k-nearest neighbor classifier was used to identify training query set and best ranking function which performs the best on this identified set is then chosen for the unseen query.

Balasubramanian and Allan (2010) proposed the "Ranker Selection" framework that predicts the difference between the effectiveness of two rankers in terms of average precision. The experiments conducted on LETOR 3.0 dataset using three rankers (RankBoost, Regression, and Frank) show that, for selecting between two rankers, a simple regression model that directly predicts differences in effectiveness, can achieve substantial improvements over the best individual ranker.

3.3.13. Query dependent loss function

Bian, Liu, Qin and Zha (2010) propose to incorporate query difference into ranking by introducing query dependent loss functions. They compare query-dependent loss function to query-dependent ranking function. According to their study, query-dependent loss function outperforms.

3.3.14. Selective pruning

Tonellotto, Macdonald and Ounis (2013) propose a novel selective framework that determines the appropriate amount of pruning aggressiveness on a per-query basis, thereby increasing overall efficiency without significantly reducing overall effectiveness. In their work, the authors aim to ensure effective and efficient retrieval, by selecting which queries should be pruned more aggressively.

4. THE SELECTIVE TERM WEIGHTING FRAMEWORK

4.1. Introduction

The previous chapter presented previously proposed selective retrieval approaches. In this section we introduce our selective term-weighting framework that selectively applies an appropriate term-weighting model from a set of candidate/representative term-weighting models based on frequency distributions of query terms.

A central concept of this framework is that the best term-weighting model, which would yield highest effectiveness, for a given unseen test query can be estimated based on the frequency distributions of the query's terms. In particular, we propose a novel query similarity, which takes into account the frequency distribution of queries' terms. This new query similarity is based on the chi-square goodness of fit test, which is used to quantify the extent to which two observed frequency distributions are similar to each other.

Our proposed approach does not require the result lists nor their associated relevancy scores returned by the test query. Since it automatically selects a term-weighting model before the actual search takes place, it is a *pre-retrieval* strategy.

The remainder of this chapter is organized as follows. Section 4.2 explains the concept of *term frequency distribution* in the context of IR and text document collections, as well as the proposed term and query similarity functions based on the chi-square goodness of fit test. Section 4.3 details the selection mechanism for selectively applying an appropriate term-weighting on a per-query basis. Section 4.4 tabulates the candidate term-weighting models. Section 4.5 discusses/highlights a limitation of the frequentist approach to IR. Section 4.6 compares the selective term-weighting framework with the existing selective retrieval approaches. Finally, a summary of this chapter is presented in Section 4.7.

4.2. Term Frequency Distribution

Frequency is how often something occurs. In context of IR and text document collections, term frequency (denoted by $\text{tf}_{t,d}$) is defined as the number of occurrences of term t in document d. This definition is referred to as *raw term frequency* or *within-document term frequency* in this dissertation.

By counting frequencies we can make a frequency distribution table. Basically we count the documents that contain the term t one time, two times, three times, and so on. According to the frequentist approach, raw term frequency is not comparable across documents because documents have varying lengths. Therefore, in this dissertation we only work with *relative term frequency*, which is calculated as raw term frequency divided by length of the document. Note that: 0 < relative term frequency ≤ 1

Dividing the within-document term frequency by the document length has following advantages: (*i*) it is equivalent to the probability that term t is chosen from the document d at random. (*ii*) in Language Modeling, it is also referred to as Maximum Likelihood Estimate (MLE) of the probability of term t under the term distribution for document d. (*iii*) relative frequencies of distinct terms in a document sum up to one. (*iv*) given that: 0 < relative term frequency ≤ 1, it is easy to create binned grouped relative term frequency distribution table of the term.

In contrast to raw term frequency, relative term frequency values are not integer. They are less than or equal to one and have many different/distinct values. Therefore, they are not suitable for creating a frequency distribution table. We have applied the following *binning approach* to tackle the problem.

4.2.1. Grouped relative term frequency distribution

We partition the interval (0-1] into 1000 bins of equal length and put relative term frequency values into these 1000 bins. Table 4.1 shows the bin intervals and the corresponding frequencies for the term *atari*. Such distribution tables are useful for

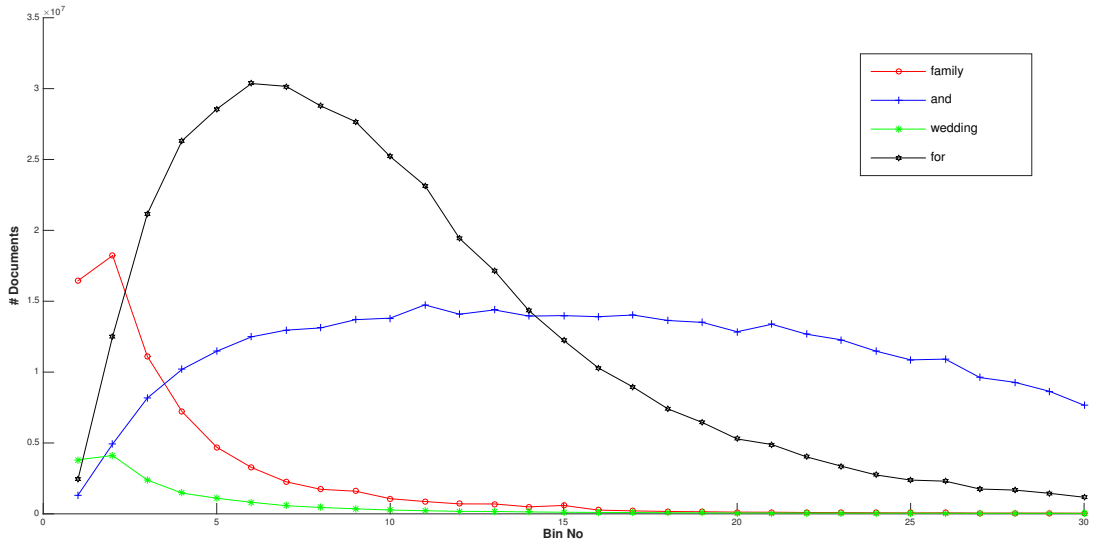

Figure 4.1. Grouped Relative Term Frequency Distribution Plot

Table 4.1. The Bin Intervals

Bin No	Bin Interval	Frequency
1	(0.000 - 0.001)	228785
2	[0.001 - 0.002)	183551
3	[0.002 - 0.003)	99254
4	[0.003 - 0.004)	63628
5	[0.004 - 0.005)	42223
6	[0.005 - 0.006)	28372
...	...	...
1000	[0.999 - 1.000]	0

comparing distribution of query terms visually. In Figure 4.1, distribution graphs of four terms (*and*, $family$, for, *wedding*) are compared. It can be observed that shapes of the distributions are quite different. Notice that $family$ and *wedding* are content bearing words, while for and *and* are function words.

4.2.2. Goodness of fit tests

Goodness of Fit (GOF) tests measure how much an observed frequency distribution differs from a theoretical distribution, such as Poisson, Hyper-Geometric, and Log-Logistic. Pearson's chi-square statistic, which is of the *nonparametric* type (Conover, 1999), is defined as follows:

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{4.1}$$

Where O_i are observed counts, E_i are corresponding expected count and n is the number of classes for which counts/frequencies are being analyzed.

χ^2 can be used to measure the discrepancy or similarity between two observed frequency distributions. Let R_i be the number of documents in bin i for the first term, S_i the number of documents in the same bin i for the second term. Then the chi-square statistic is

$$\chi^2 = \sum_{i=1}^{n} \frac{(R_i - S_i)^2}{R_i + S_i} \tag{4.2}$$

4.2.3. Term similarity based on frequency distributions

Once grouped relative term frequency distribution/table of a term is obtained, it is straightforward to quantify its similarity to another term by using χ^2 GOF statistics given by the Equation 4.2. Table 4.2 presents grouped relative term frequency distribution table of different words/terms. Different from the Table 4.1, this time we include the *zeroth* bin, which is the number of documents that do *not* contain the term. By doing so, we take into account the IDF effect of the term, and all rows sum up to the same value: total number of documents in the collection. This also satisfies the *unequal number of data points* requirement in χ^2 calculation. Notice that Equation 4.2 apply to the case where the total number of data points (document frequency) is the same in the two binned sets. (e.g. $\sum_{i=1}^{n} R_i = \sum_{i=1}^{n} S_i$)

Since different terms may have different document frequency (DF) values/ranges, without inclusion of the zeroth bin, we would not directly use the Equation 4.2 for terms having different DF values/ranges.

Although χ^2 was originally designed to quantify how much an observed frequency distribution is representative of a theoretical distribution it can also be used

Table 4.2. Grouped Relative Term Frequency Distribution Table

Term \ Bin No	0	1	2	3	4	5	6	7
warren	499902875	1698022	1045982	461024	247844	150566	90134	58673
yahoo	483767215	4847226	5968976	3125663	1678247	1034722	677554	473663
diversity	498264010	2263737	1415938	716437	403022	239535	149181	98598
euclid	503552774	118475	75725	45912	28992	18409	12270	8443
fish	489620024	4607778	3526704	1829877	1040154	690274	460965	329652
poker	497452573	1336556	1346682	758706	508246	356959	238320	167325
the	60222095	1156860	4264509	6727695	8351399	9374867	10152057	10202091

to measure the similarity between two observed (grouped relative term) frequency distributions, where there are no expected values but two observed values. Let R_i be the number of documents in bin i for the first term R, S_i the number of documents in the same bin i for the second term S. Then the term similarity based on the chi-square statistic is

$$\chi^2 = \sum_{i=0}^{n} \frac{(R_i - S_i)^2}{R_i + S_i} \tag{4.3}$$

where n is the number of bins which is 1000 in the present dissertation. It should be noted that we have borrowed the Equation 4.2 from the book by Press, Teukolsky, Vetterling and Flannery (2007).

We further divide each bin value entry of the table by the total number of documents in the collection to make these distribution values collection independent. This normalization/standardization does not affect relative order of χ^2 values of terms that belong to the same collection (*within-collection*), but it is useful/necessary for *cross-collection* experiments (test and train queries' term frequency distributions come/extracted from different collections).

The order of terms in χ^2 is not important, i.e., it is symmetric/commutative: $\chi^2(lymphoma, paralegal) = \chi^2(paralegal, lymphoma)$. The commutative property of χ^2 makes the calculation efficient cacheable in an actual real-world IR system. High values of χ^2 implies a poor fit between terms, while zero value represents a perfect fit. Therefore, similarity of the same two terms is always zero. For example: $\chi^2(yahoo, yahoo) = 0$.

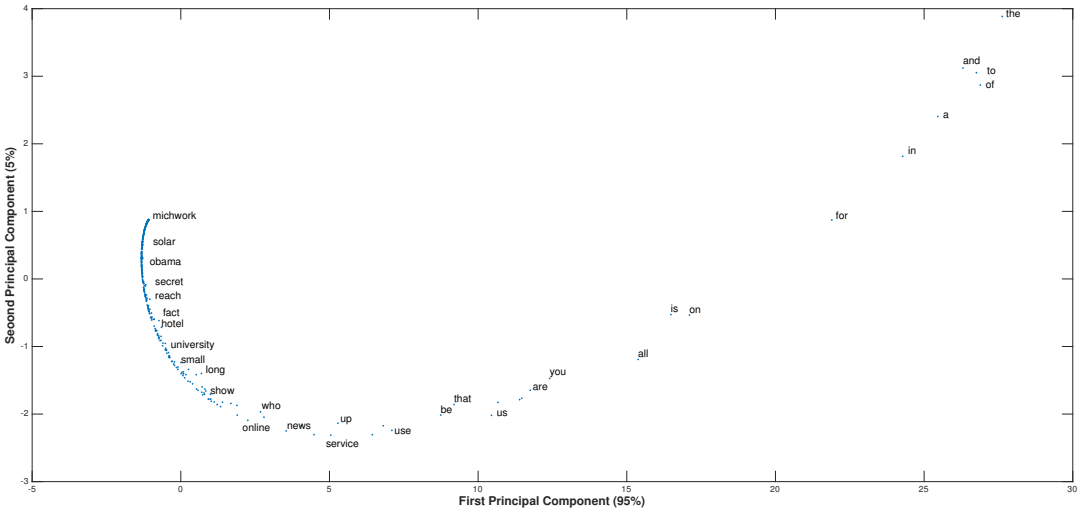

Figure 4.2. Multidimensional Scaling Analysis of ClueWeb09 Query Terms

We take a matrix of χ^2 similarities between query terms from the ClueWeb09 dataset. Multidimensional scaling (MDS) analysis of this matrix is depicted on Figure 4.2, which visualizes the level of χ^2 similarity of individual query terms by reproducing the similarities based on two dimensions. As a result of the MDS analysis, we obtain a two-dimensional representation of the χ^2 similarities of the terms. Total 411 distinct query terms, some of whose text are made visible, are depicted in the Figure 4.2. As can be seen, the term *the*, is at the upper right corner, is the most radical/distant term. The rarest term *michworks*, observed only in 249 documents, falls in the upper left corner.

4.2.4. Query similarity based on frequency distributions

It should be noted that previously described grouped frequencies are calculated for a single term. However, in practice, a query $Q = \{t_1, t_2, \ldots, t_n\}$ can comprised of multiple terms. Therefore the Equation 4.3 cannot be directly used for a query to query similarity. As a remedy to the problem we propose two approaches: (i) Average and (ii) Cartesian.

Average query similarity It is worthwhile to note that, some of the learning to rank (LETOR) features are based on terms. To obtained a query feature from

Table 4.3. Query Frequency Distribution

Term \ Bin No	0	1	2	3	4	5	6	7
obama	492284836	2573539	3164159	1807392	1048292	681580	466128	337461
family	430734219	16443075	18223427	11111546	7213982	4699781	3273734	2261226
tree	482831202	6754088	5385276	2719087	1564855	1005215	690448	479980
Average	468616752	8590234	8924287	5212675	3275710	2128859	1476770	1026222

Table 4.4. Initial Cartesian Table: filled with all pairs of $\chi^2(x,y)$

	air	**travel**	**information**
internet	0.163	0.012	0.001
phone	0.006	0.220	0.145
service	0.148	0.014	0.002

multiple term features, an aggregation strategy is necessary. For example, the final value of IDF in LETOR 3.0 dataset is the sum of the IDF of each query term: $idf(Q) = \sum_{t \epsilon Q} idf(t)$. In a similar fashion, to aggregate multiple binned frequency distributions, we simply take average of the frequencies of bin i.

Table 4.3 demonstrates the procedure for the query *obama family tree*. At the last row, three term frequencies are averaged for each bin. Thus, we obtain a pseudo frequency distribution for a *query*, as if it is a single term. This transformation allows us to calculate chi-square statics of two queries, where queries were treated as terms.

Cartesian query similarity In this novel approach, we do not aggregate frequency distributions of different terms of a query into one. Instead, we employ cartesian product of query terms. We consider all possible pairs of terms, which are member of two queries.

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the first query comprised of n terms, $Y = \{y_1, y_2, \ldots, y_n\}$ be the second query comprised of n terms. A cartesian product table $X \times Y = \{\chi^2(x,y) \| x \in X \wedge y \in Y\}$ is constructed from all pairs of (x,y) where $y \in Y$ and $x \in X$. Table 4.4 presents such table constructed for $X = \{internet, phone, service\}$ and $Y = \{air, travel, information\}$.

Next we find the minimum element (strongest matching pair) in the table,

Table 4.5. Remaining Cartesian Table: after the first pair's match

	air	**travel**
phone	0.006	0.220
service	0.148	0.014

which is $\chi^2(internet,information)$=0.001, to determine the first pair. Once a pair is determined (a term in X finds its significant other in Y), we remove the whole column and row that form the pair from the table and repeat the process until the table is empty. Table 4.5 shows the remaining entries after the removal of the first couple. The smallest entry among the remaining entries forms the second couple, which is $\chi^2(phone,air)$=0.006. Thus, our third and last couple becomes $\chi^2(service,travel)$=0.014. Finally we use the normalized Euclidean distance of all three couples as a resulting query similarity.

$$sim(X,Y) = \frac{\sqrt{0.001^2 + 0.006^2 + 0.014^2}}{3} = 0.005 \tag{4.4}$$

In this similarity algorithm, every term finds its unique couple. That why we name this similarity as *CoupleSimilarity*. However, this similarity requires that the queries to be compared must have equal lengths. Next we describe how we handle unequal query lengths.

When the lengths of the queries are equal, we apply CoupleSimilarity. Elsewhere, we label the queries as Q_{long} and Q_{short} according to their query lengths: number of terms in a query. We generate Q_{short} combination of Q_{long} and call CoupleSimilarity for each piece, which has the same length as Q_{short}. For example for $X = \{internet, phone, service\}$ and $Y = \{disneyland, hotel\}$ we obtain $\binom{Q_{long}}{Q_{short}} = \binom{3}{2} = 3$ pieces for the long query X: [internet, phone] [internet, service] [phone, service]. We invoke CoupleSimilarity for each piece using Q_{short} :

- CoupleSimilarity(*disneyland hotel*, *internet phone*)
- CoupleSimilarity(*disneyland hotel*, *internet service*)

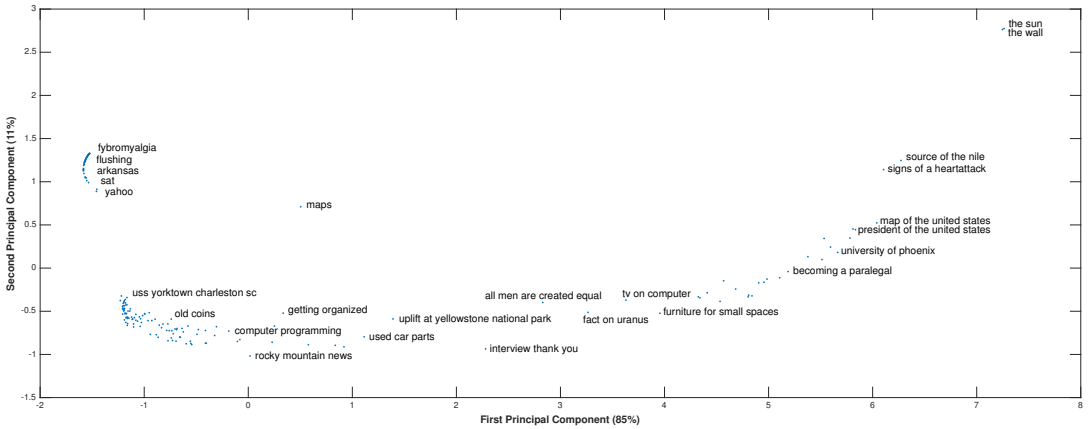

Figure 4.3. Multidimensional Scaling Analysis of ClueWeb09 Queries

- CoupleSimilarity(*disneyland hotel*, *phone service*)

To obtain the final similarity score, we take the average of the minimum and the maximum of the list: $similarity = \frac{max(list)+min(list)}{2}$.

We take a matrix of CartesianSimilarity scores between queries from the ClueWeb09 dataset. MDS analysis of this matrix is depicted on Figure 4.3, which visualizes the level of CartesianSimilarity of individual queries by reproducing the similarities based on two dimensions. As a result of the MDS analysis, we obtain a two-dimensional representation of the CartesianSimilarity scores of the queries. There are total 188 queries, some of whose text are visible, depicted in the Figure 4.3. As can be seen, the queries having common term (e.g., *the*, *of*, *a*) are grouped at the upper right corner. By their very nature, one-term queries are usually comprised of specific terms, since there is only one term in them. One-term queries are positioned at the upper left corner, with the exception of the query *maps*, which is neither specific nor common. The queries differ only in one term are also very close to each other in the Figure 4.3.

Both AverageSimilarity and CartesianSimilarity satisfy following properties:

- Commutative: $\text{sim}(X,Y) = \text{sim}(Y,X)$
- $\text{sim}(X,X) = 0$

- $\text{sim}(X,Y) \geq 0$

4.3. Selection Mechanism

Based on the query similarity implementations introduced in the previous section, our term-weighting model selection mechanism can be summarized as follows:

- Initially, on a training collection, we have a set queries $Q = \{q_1, q_2, q_3, \ldots, q_n\}$ and a set of candidate term-weighting models $M = \{$BM25, DFIC, DFRee, DLH13, DPH, DirichletLM, LGD, PL2$\}$
- For each term-weighting model $m_i \in M$, we create a *winner* list, whose elements are the queries that the model m_i attained the *highest* effectiveness score.
- For each term-weighting model $m_i \in M$, we create a *loser* list, whose elements are the queries that the model m_i attained the *lowest* effectiveness score.
- For a given unseen test query q_t, we calculate a *similarity* distance to every term-weighting model $m_i \in M$. The distance is the average of the sums of a query similarities between the test query q_t and winner list of the term-weighting model $m_i \in M$.
- For a given unseen test query q_t, we calculate a *dissimilarity* distance to every term-weighting model $m_i \in M$. The distance is the average of the sums of a query similarities between the test query q_t and loser list of the term-weighting model $m_i \in M$.
- We select the model $m_i \in M$ whose winner list is the closest to the test query q_t and whose loser list is farthest away from the test query q_t.

It is worthwhile to note that winner/loser list depends on the target effectiveness metric (e.g. NDCG@100, MAP, etc) that we want to optimize. In other

words, winner/loser list of a term-weighting model can change metric to metric. We choose to optimize NDCG@100 and MAP, because they both asses overall result quality.

The selective term-weighting algorithm, written in pseudo-code, can also be found in Algorithm 1 in which previously described query similarity is denoted by similarity(q_t, q_j).

4.4. Candidate Term-Weighting Models

In the present dissertation, eight term-weighting models are used to choose from. Each term-weighting model assumes a distribution for term frequencies on document collections. The models and distribution assumed by them are listed in Table 4.6. Note that DFIC is *nonparametric*, which means it does not make any assumptions about the underlying term frequency distribution.

4.5. Frequentist Approach to Information Retrieval

The fundamental assumption of the frequentist approach to IR is that the more a document contains a term t, the more the document treats the term t. This suggest that observed frequency (distribution) of a term t should be different on relevant documents than that of non-relevant documents. However, relevancy cannot be explained solely by term frequency. There exists other factors such as spam, document quality (PageRank, HitRank), recency, etc. The extend that this assumption is valid varies across queries; some queries satisfy it well while other queries satisfy this assumption more loosely. It is easy to show an counter example where this assumption does not hold. Figure 4.4 shows relative term frequency distribution of the query $fybromyalgia$ over query relevance judgments. The x-axis shows relevance grades (4, 2, 1, 0, -2), in which grades 1/2/3/4 are treated as relevant and grades 0/-2 as non-relevant. As can be observed from the figure relevant and non-relevant documents cannot be distinguished. In other words, the query term $fybromyalgia$ tends to have same distribution over relevant and non-relevant docu-

for $m_i \in M$ **do**
    create winner list for m_i;
    create loser list for m_i;
end
Let q_t be the unseen test query;
for $m_i \in M$ **do**
    acc $\leftarrow$ 0;
    c $\leftarrow$ 0;
    for $q_j \in$ *winner list of* m_i **do**
        if $q_t \neq q_j$ **then**
            acc $\leftarrow$ acc + similarity(q_t, q_j);
            c $\leftarrow$ c+ 1;
        end
    end
    m_i.`similarity` $\leftarrow$ acc / c;
    acc $\leftarrow$ 0;
    c $\leftarrow$ 0;
    for $q_j \in$ *loser list of* m_i **do**
        if $q_t \neq q_j$ **then**
            acc $\leftarrow$ acc + similarity(q_t, q_j);
            c $\leftarrow$ c+ 1;
        end
    end
    m_i.`dissimilarity` $\leftarrow$ acc / c;
end
return $\arg\max_{m_i \in M} f(m_i) = \{m_i \mid m_i.\texttt{dissimilarity}/m_i.\texttt{similarity}\}$;

Algorithm 1: The selective term-weighting framework

Table 4.6. Participant weighting models and their underlying distributions

No	Model	Distribution	Reference
1	BM25	Poisson Distribution	Robertson and Zaragoza (2009)
2	Dirichlet	Binomial/Multinomial Distribution	Zhai and Lafferty (2004)
3	DFIC	Chi-Squared Distance	Kocabaş, Dinçer and Karaoğlan (2014)
4	DFRee	Hypergeometric Distribution	Amati (2009)
5	DLH13	Hypergeometric Distribution	Amati (2006)
6	DPH	Hypergeometric Distribution	Amati (2006)
7	LGD	Log-Logistic Distribution	Clinchant and Gaussier (2010, 2011)
8	PL2	Poisson Distribution	Amati and Van Rijsbergen (2002)

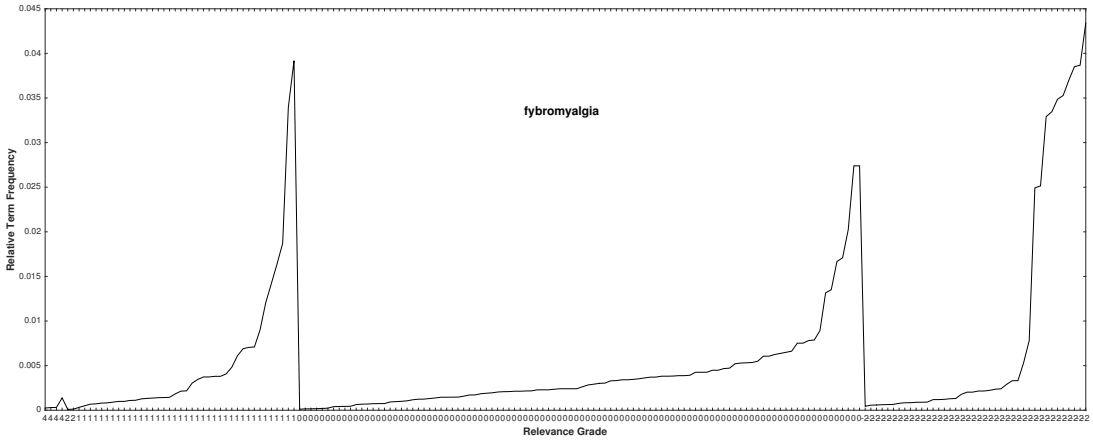

Figure 4.4. Distribution of *fybromyalgia* over Relevance Judgments.

ments. This example clearly demonstrates that the frequentist approach to IR has its limits. However, to some extend, this assumption holds, especially in controlled datasets, because, if not all, most of the term-weighting models are built upon this.

4.6. Discussion

The proposed selective term-weighting framework is different from the previously proposed selective retrieval approaches (Chapter 3) in several aspects. First, most of the studies employ k-NN algorithm and report results for the best k value. It is not clear what and how to set k in a real-world application. By contrast, our proposed approach is parameter-free.

The other major difference is that we leverage the information carried by the queries that the retrieval strategy performed the worst in our selection mechanism. To the best of our knowledge, existing studies consider only the queries that the retrieval strategy performed the best. Actually, the ultimate task in selective IR is not a plain classification task, but is a ranking task: retrieval strategies are ranked for a given query. Thus, an ideal training procedure should take into account the per-query relative rankings of the retrieval strategies and the magnitude of the effectiveness scores that create the differences/ranks.

4.7. Summary

In this chapter, we describe the concept of *term frequency distribution* from IR perspective. We drive term and query similarity methods based of the goodness of fit between frequency distributions of query terms. We illustrate workings of our query similarity function on an example. We explain how a term-weighting model can be selected from the candidate set for a given query using the similarity methods. We call attention to limitations of the frequentist approach to IR.

5. EXPERIMENTAL METHODOLOGY

5.1. Introduction

The previous chapter presented a novel selective term-weighting approach/framework for selectively applying an appropriate term-weighting model for a given query. This chapter describes the methodology we adopted for the experimental evaluation of the proposed selective term-weighting approach/framework. We describe the datasets, the corresponding query sets, spam filtering strategy, the evaluation measures used for the analysis, the baseline retrieval models, and the retrieval engine.

5.2. Datasets

For the empirical analysis, we used a large number of standard/representative TREC Web collections of varied sizes and contents. In particular, we used newly released ClueWeb{09|12} corpora as suggested by Metzler and Kurland (2012).

5.2.1. ClueWeb09-A

The ClueWeb09A full collection consists of roughly 1 billion Web pages, comprising approximately 25TB of uncompressed data (5TB compressed) in multiple languages. The dataset was crawled from the Web during January and February 2009. In our experiments, we use English subset of ClueWeb09, which consists of all the English pages (over 500 million) in the dataset.

For ClueWeb09 dataset, a total of 200 information needs with relevance judgments are released during TREC Web Tracks (WT) which ran from 2009 to 2012. However, we excluded three topics: 20 from Web Track 2009, 95 and 100 from Web Track 2010. Because, for topic 20, none of the participating runs had returned any relevant document. Relevance judgements do not exists for topics 95 and 100. Thus, there remains 197 valid topics for ClueWeb09 dataset.

5.2.2. ClueWeb09-B

ClueWeb09B corpus, which is "Category B" portion of ClueWeb09A, comprises the first 50 million English-language pages of the full dataset, including the entirety of the English-language Wikipedia. ClueWeb09B and ClueWeb09A use the same topic set. Further information on the ClueWeb09 collection may be found on the Lemur project website[1].

5.2.3. Million query 2009

TREC 2009 Million Query (MQ09) Track[2] investigates whether it is better to evaluate using many shallow judgments or fewer thorough judgments and whether small sets of judgments are reusable.

The MQ09 collection contains 561 queries, first 50 of which were taken from Web Track 2009 (WT09). Rest of the queries has shallow relevance judgements. Query relevance judgments of MQ09 is published as five-column `prels` file instead of four-column `qrels` file. Therefore, `statAP_MQ_eval_v4.pl`[3] evaluation script is used for MQ09 evaluation. To stay as close to NIST data as possible, we report the statistical average precision (statAP), which was one of the official metrics for the million query track (Carterette et al., 2009). Note that, MQ09 uses ClueWeb09B as the document collection.

5.2.4. ClueWeb12-B13

The ClueWeb12 full dataset consists of 733,019,372 English Web pages, collected between February 10, 2012 and May 10, 2012. ClueWeb12 is a companion or successor to the ClueWeb09 Web dataset. For ClueWeb12 dataset, a total of 100 information needs with relevance judgments are released for Web Track 2013 and 2014.

ClueWeb12-B13 dataset is created by taking every 14^{th} document (WARC

[1] `http://lemurproject.org/clueweb09`
[2] `http://ir.cis.udel.edu/million/data.html`
[3] `http://ir.cis.udel.edu/million/statAP_MQ_eval_v4.pl`

Table 5.1. Statistics of the TREC Datasets (as indexed by Apache Lucene)

Collection	# Tokens	# Documents	Avg. Length
CW09B (NoAnchor)	39,227,244,424	50,220,154	781.1
CW09B (AnchorText)	51,928,477,704	50,220,293	1034.0
CW09A (NoAnchor)	331,995,474,761	503,892,054	658.9
CW09A (AnchorText)	366,636,877,415	503,896,369	727.6
CW12B (NoAnchor)	37,254,681,357	52,238,715	713.2
CW12B (AnchorText)	37,919,236,268	52,244,050	725.8

response record), from each of the files of the full dataset. Therefore, it is a representative or uniform 7% sample of the full dataset. Further information on the ClueWeb12 collection may be found on the Lemur project website[4].

5.2.5. Summary

After we strip Hyper Text Markup Language (HTML) tags using jsoup[5] library (version 1.8.3), we consider HTML documents as a whole and index entire document. We do not employ different document representations (URL, title, body, keywords, description, etc.). However, we have built an additional index in which anchor texts from in-links are treated as part of the documents. Therefore, we have built two indexes with the following tags: `AnchorText` and `NoAnchor`, in which documents are indexed with and without anchor text respectively.

Table 5.1 reports the total number of indexed documents/tokens and the average document length for each dataset. HTML tag stripping procedure produced/yielded an empty string for some documents, which are skipped during indexing. That's why the number of documents are slightly different for `AnchorText` and `NoAnchor` indexes of a dataset. This implies that `AnchorText` index contains a few documents that are composed of anchor texts only. Statistics reported in Table 5.1 depend on both the tokenization and HTML cleaning/parsing strategies. For example, Kulkarni and Callan (2015) reported average document length of 918 for the CW09B dataset, which is larger than that is reported in this dissertation.

[4] `http://lemurproject.org/clueweb12`
[5] `http://jsoup.org`

Listing 5.1. Example of Information Needs (Queries)

```
137: rock and gem shows
138: jax chemical company
139: rocky mountain news
140: east ridge high school
301: International Organized Crime
```

Table 5.2. Statistics of Query Sets

Track	# Queries	Average Query Length	Average # Relevant Documents per Query	Average # Non-Relevant Documents per Query	# Relevance Levels
MQ09	561	2.6	15.2 (± 25.7)	37.9 (± 48.1)	3
WT09	49	2.1	139.7 (± 78.8)	332.9 (± 79.9)	3
WT10	48	2.0	108.9 (± 70.7)	417.8 (± 132.1)	5
WT11	50	3.4	63.0 (± 63.5)	323.6 (± 101.3)	5
WT12	50	2.3	70.3 (± 55.2)	249.8 (± 87.1)	6
WT13	50	3.3	82.7 (± 64.8)	205.8 (± 100.2)	6
WT14	50	3.3	113.1 (± 74.8)	174.6 (± 81.3)	6

5.3. Topics - Queries

In our experiments, we discard the topics that have no relevant documents in the judgment set: we only work with *valid* topics. Exact number of topics used in this dissertation is 1,107 and the statistics for these queries and their corresponding relevance judgements are provided in Table 5.2. All the collections have graded relevance assessment at minimum of three-point scale. We used the initial release of topics which included only the query/title field, as shown in the Listing 5.1.

5.4. Baselines

In this section, we present two different baseline families.

5.4.1. State-of-the-art term-weighting models

We have compared the effectiveness of the proposed selective term-weighting scheme with eight state-of-the-art retrieval models.

- BM25: representative of the classical probabilistic model.
- Dirichlet: representative of the language model family.

- PL2: representative of the divergence from randomness framework.
- DPH: representative of hyper-geometric models of IR.
- DLH13: representative of hyper-geometric models of IR.
- DFRee: representative of parameter-free models of IR.
- DFIC: representative of nonparametric models of IR.
- LGD: representative of information based models of IR.

We choose Dirichlet (Zhai and Lafferty, 2004) smoothing from language models, because it is known to be the most effective among the language models (Croft et al., 2009). Hence, our set of baselines contains members from all state-of-the-art retrieval families.

5.4.2. State-of-the-art selective term-weighting

To the best of our knowledge, He and Ounis (2003, 2004b) were the first to carry out research on ***selective term-weighting*** among several selective retrieval approaches. They defined the problem as follows:

> Many term-weighting models have been proposed for information retrieval. For **a given collection** and **a given query**, it is an interesting and challenging problem to automatically select **the best term-weighting model**, which would provide the best retrieval effectiveness. This problem is referred to as the ***model selection*** problem.

Indeed, the term *model* is a very generic word. In the field of machine learning, the term model is used to describe what is obtained after training phase of a learning algorithm. However, in the context of the present dissertation, model means *term-weighting scheme* or *weighting function*. It is not a model that is learnt, however, but rather it is a static function/formula of term and corpus

statistics (e.g. $tf \cdot idf$). Therefore, it must not be confused with the model that is trained/learned/estimated by machine learning (learning to rank) techniques.

He and Ounis (2004b) proposed a query-based pre-retrieval approach which automatically selects the best-performing retrieval model before the retrieval process takes place. Thus, they motivated their research as a *pre-retrieval* remedy to the *poorly performing* queries.

As noted by He and Ounis, there were previously proposed approaches (Jin et al., 2001; Manmatha et al., 2001; Si and Callan, 2002) on the model selection problem. However, these approaches are *post-retrieval* approaches, which requires analysis of the result list and relevance scores. In other words, *post-retrieval* approaches cannot select the optimal model prior to the retrieval process. For this reason, post-retrieval approaches are out of the scope of the present dissertation. Analyzing and merging the result lists returned by different (heterogeneous) search engines/systems into a single, coherent ranked list is covered by a research area known as *Distributed Information Retrieval* (Callan, 2000) or *Federated Search* (Shokouhi and Si, 2011).

He and Ounis cluster the queries according to their intrinsic features and associate the best-performing term-weighting model to each cluster. They used 11 different DFR term-weighting models to choose from. Their results show that query-based model selection approach does improve the poorly-performing queries compared to a baseline where a unique retrieval model is applied indifferently to all queries.

Queries as feature vectors He and Ounis (2003, 2004b) proposed following three factors for the feature vector of query:

1. The query length (ql) is the number of unique terms in the query.
2. The relative informative amount carried in each query term (γ) is defined as the quotient of the minimum IDF divided by the maximum IDF among the

query terms:

$$\gamma = \frac{IDF_{min}}{IDF_{max}}, \tag{5.1}$$

where $IDF(t) = \log(\frac{N}{df(t)})$

3. The clarity/ambiguity of a query (ω) is measures as:

$$\omega = -\frac{\log(n_Q/N)}{\log N}, \tag{5.2}$$

where n_Q is the number of documents containing at least one of the query terms and N is the number of documents in the collection.

He and Ounis (2003, 2004b) represent each query by the feature vector qf given as:

$$\overrightarrow{qf} = (\rho \cdot ql, \gamma, \omega), \tag{5.3}$$

where ρ is a parameter that was set to 0.2 by He and Ounis. He and Ounis (2004b) adopt the CURE algorithm (Guha et al., 1998) to cluster the feature vectors in the above three-dimensional space. Initially, each vector is an independent cluster. If there are n vectors to be processed, algorithm starts with n clusters. The similarity between two clusters is measured by the cosine similarity of the two closest vectors (having the highest cosine similarity), where the two vectors come from each cluster respectively. Then, we merge the closest pair of clusters (according to the cosine similarity measure) as a single cluster. The merging process is repeated until it results in k clusters. Hence, the number k of clusters is the halting criterion of the algorithm.

The model selection mechanism A set of training queries are clustered according to their features. For each cluster, the best-performing model is identified in terms of the effectiveness measures. Then an unseen query is served with the best-performing model associated with the closest cluster to the query. In our ex-

periments, we adopted the approach proposed by He and Ounis (2003, 2004b) and used its results as a selective term-weighting baseline.

5.5. Effectiveness Measures and Evaluation Tools

A set of standard IR effectiveness metrics, NDCG@100 and MAP@1000, was used to measure retrieval effectiveness at various cut-off levels in the present dissertation. NDCG@k can leverage graded relevance, while MAP is a standard metric for binary relevance assessments.

We used `gdeval.pl` (version 1.3) TREC evaluation tool (downloaded from trec-web-2014[6] GitHub repository) to calculate NDCG@k values reported in this dissertation. In order to measure the MAP (based on a maximum of 1000 retrieved documents), we used the `trec_eval`[7] utility (version 9.0), which is the standard tool used by the TREC community.

Another important detail is that we have always used query relevance judgements published for full datasets (Category A) in our experiments. In other words, to make reported effectiveness values comparable to other/existing studies, we evaluate result lists obtained from category B subset using query relevance judgements obtained from full dataset (Category A).

5.6. Optimization of the Free Parameters

All the baseline models (except DFIC, DLH13, DPH, DFRee) contain one or more free parameters. It is very important to tune these parameters properly because they affect the effectiveness to a statistically significant degree. For the sake of reliable and fair comparison, we use best parameters values that attained the highest evaluation metric for each dataset.

For BM25, we borrowed the intervals from Lv and Zhai (2012): b from 0.1 to 0.9 in increments of 0.1 and k_1 from 0.2 to 3.0 in increments of 0.2. Table 5.3

[6] `http://github.com/trec-web/trec-web-2014`
[7] `http://trec.nist.gov/trec_eval/trec_eval_latest.tar.gz`

Table 5.3. Free parameter values

Model	Parameter & Set of Values
BM25	$k_1 \in \{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0\}$
BM25	$b \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
PL2, LGD	$c \in \{0.25, 0.5, 0.8, 1, 2, 3, 5, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30\}$
Dirichlet	$\mu \in \{10, 50, 100, 200, 500, 800, 1000, 1500, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000\}$

Table 5.4. Trained free-parameter values of `NoAnchor` index

		BM25	LGD	PL2	Dirichlet
CW09A	NDCG100	k_1=1.0 b=0.4	c=2.0	c=3.0	μ=500
CW09A	MAP	k_1=1.2 b=0.3	c=2.0	c=8.0	μ=500
CW09B	NDCG100	k_1=1.2 b=0.2	c=18.0	c=18.0	μ=2000
CW09B	MAP	k_1=2.0 b=0.2	c=8.0	c=12.0	μ=1500
CW12B	NDCG100	k_1=1.8 b=0.2	c=5.0	c=10.0	μ=2000
CW12B	MAP	k_1=1.4 b=0.2	c=5.0	c=5.0	μ=1500
MQ09	NDCG100	k_1=1.6 b=0.5	c=2.0	c=3.0	μ=500
MQ09	statMAP	k_1=1.4 b=0.3	c=5.0	c=8.0	μ=800

shows parameter ranges used during parameter tuning for different free-parameters of retrieval models.

Trained free-parameter values are given in Table 5.4 for `KStem` index of datasets. It is worthwhile to note that optimum value of a model's free-parameter varies by dataset and by effectiveness measure.

5.7. Spam Filtering

Cormack, Smucker and Clarke (2011) carried out the first systematic study of spam in the English ClueWeb09 (category A) dataset and presented the first quantitative results of the impact of spam filtering on IR effectiveness. They reported that a substantial fraction of ClueWeb09 are spam and the use of spam filtering significantly improves retrieval effectiveness for most of the systems that participated

Table 5.5. Spam threshold t% values that maximized the mean effectiveness of eight term-weighting models

	ClueWeb09A		ClueWeb09B		MillionQuery09	
	NoAnchor	Anchor	NoAnchor	Anchor	NoAnchor	Anchor
NDCG@100	55	55	15	10	10	10
MAP	50	20	15	10	10	0

Listing 5.2. Excerpt from ClueWeb09 spam Fusion scores

```
36 clueweb09-en0000-00-00000
41 clueweb09-en0000-00-00001
44 clueweb09-en0000-00-00002
60 clueweb09-en0000-00-00003
61 clueweb09-en0000-00-00004
```

in the TREC 2009 Web Track. Cormack et al. generated following four different rankings of the spamminess of English documents in the ClueWeb09 dataset and made them publicly available to other researchers in the two-column format shown in Listing 5.2. The second column is the official document identifier, whereas the first column is percentile score that indicates the percentage of the documents in the corpus that are "spammier." That is, the spammiest 10% of the documents have percentile score >10.

- UK2006: A set of labels trained against a small set of Web pages containing 746 spam pages and 7,474 non-spam pages.
- Britney: Derived from results returned for popular queries given to commercial search engines.
- Group X: Manually labelled from results for queries from the 2009 TREC Ad-hoc task.
- Fusion: A combination of the other three methods.

In our experimentations, we employed the *fusion* spam scores to exclude the spammiest $t\% \in \{0, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90\}$ pages from the result lists of the term-weighting models. To make sure there remains at least 1000 documents after the percolation process, initially we fetch $10 \times 1000 = 10,000$ documents per query. If no documents are left for a query, we return the single document with the ID of "clueweb09-en0000-00-00000" at rank one, which represents zero documents for the ClueWeb09 dataset. Doing so maintains consistent evaluation results (averages over the same number of queries) and does

not break evaluation tools. Figure 5.1 shows NDCG@100 as a function of t% for each of the term-weighting models on the ClueWeb09A dataset. We use the spam threshold t% setting that maximizes the mean retrieval effectiveness of eight term-weighting models. That is, t=55% in the Figure 5.1.

The same methodology is repeated for the ClueWeb12-B13 dataset. However, unlike the ClueWeb09, the mean retrieval effectiveness of eight term-weighting models degraded with the spam filtering performed for the ClueWeb12-B13 dataset. In others words, mean effectiveness of eight models is maximized at t=0% (e.g. no spam filtering), which holds true for all effectiveness measures. This is expected because during crawling of ClueWeb12, a blacklist[8] was used to avoid sites that are reported to distribute pornography, malware, and other material that would not be useful in a dataset intended to support a broad range of research on information retrieval and natural language understanding (Callan, 2012). By contrast, ClueWeb09 intended to provide the real Web to researchers by means of unfiltered content (Callan et al., 2009).

As can be seen from Table 5.5, best spam threshold t% setting varies by dataset and by effectiveness measure.

5.8. Apache Lucene

Apache Lucene (Białecki et al., 2012) is used as a retrieval engine in our experiments. We adopted several term-weighting model implementations from Terrier[9] (version 4.0) retrieval platform to Lucene[10] (version 5.4.0).

Preprocessing We keep the preprocessing of documents and queries minimum: we used KStemming (Krovetz, 1993), which is less aggressive than Porter's, in our experiments. Following the rationale of Fang et al. (2011), we do not perform stop word removal because stop words are essential for certain queries such as: "*to be or*

[8] `http://urlblacklist.com`
[9] `http://terrier.org`
[10] `http://lucene.apache.org`

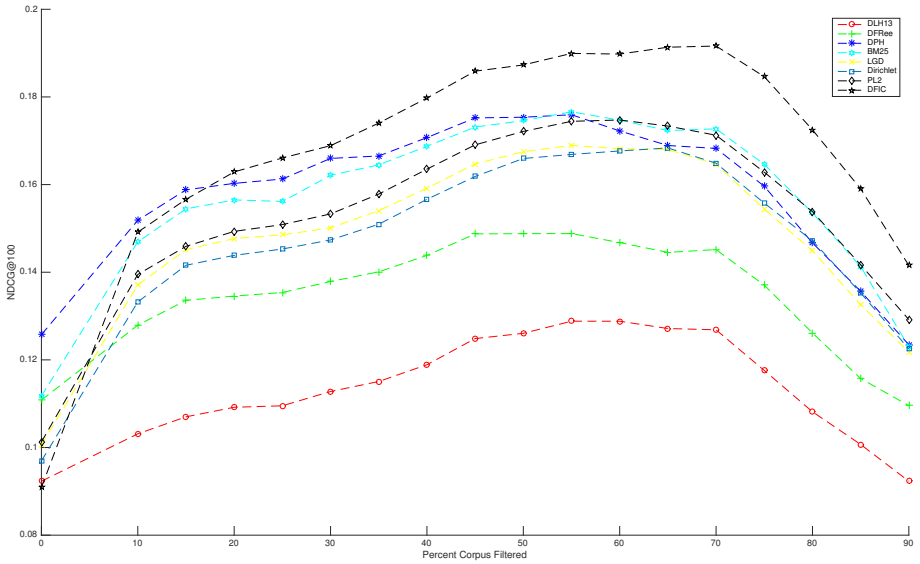

Figure 5.1. Effect of spam filtering on the effectiveness of eight term-weighting models. Effectiveness is shown as NDCG at 100 documents returned (NDCG@100) as a function of the fraction of the ClueWeb09A corpus that is labeled spam.

not to be," "*the current*," "*the wall*," "*the who*," and "*the sun*." Yet, a truly robust retrieval model should be able to cope with stop words automatically. Thus, our resulting preprocessing pipeline filters `ClassicTokenizer` with `ClassicFilter`, `LowerCaseFilter` and `KStemFilter`.

In the Lucene ecosystem, text analysis comprises a process where plain text is converted into a stream of tokens by tokenization, lowercasing, ASCII-folding, stemming, synonym expansion, stop word removal, etc. An analyzer, which encapsulates text analysis, comprise three parts: (*i*) zero or more char filters, (*ii*) a single tokenizer, and (*iii*) zero or more token filters. For each part, Apache Lucene provides a selection with several built-in implementations. For instance, KStemFilter can be used to reduce words to their stems.

The text analysis components can be chained together to create complex analysis pipes, which apply a series of transformations to each token. However, the order of the components is of crucial importance. For example, all of the terms must already be in lowercase for the KStemFilter (which is responsible for

CT	What	is	the	country's	biggest	coal	producing	state
CF	What	is	the	country	biggest	coal	producing	state
LCF	what	is	the	country	biggest	coal	producing	state
KSF	what	is	the	country	biggest	coal	produce	state

Figure 5.2. Sample Text Analysis

stemming) to work correctly. In other words, the input must be lowercased by an upstream component such as LowerCaseFilter or LowerCaseTokenizer.

Figure 5.2 visualizes every analysis step performed on the sample text "*What is the country's biggest coal-producing state?*" using our preprocessing pipeline. It should be noted that the sample text is first parsed with the ClassicTokenizer (CT) and then each token passes through ClassicFilter (CF), LowerCaseFilter (LCF), and finally KStemFilter (KSF).

5.9. Conclusions

Reproducibility has recently attracted attention from the IR community.

- The Reproducible IR Research Track at ECIR 2015 (Hanbury et al., 2016) & ECIR 2016 (Ferro et al., 2016)
- The Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR) at SIGIR 2015 (Arguello et al., 2015, 2016)
- The Open-Source IR Reproducibility Challenge (Lin et al., 2016)
- The Open Runs initiative introduced in TREC 2015 (Voorhees et al., 2016)

Above dedicated tracks and workshops are the evidences. This new line of research focuses on the repeatability, reproducibility, and generalizability of previously published methods and results.

- Repeatability: repeating a previous result under the original conditions (e.g., same dataset and system configuration)

- Reproducibility: reproducing a previous result under different, but comparable conditions (e.g., different, but comparable dataset)

- Generalizability: applying an existing, empirically validated technique to a different IR task/domain than the original

Following the trend, this chapter provided every detail of our experimental setup necessary to successfully repeat and reproduce the experiments, using Apache Lucene (`https://lucene.apache.org`).

To further promote open-source sharing, repeatability and reproducibility, we will publish our source code on GitHub, in the public repository (`https://github.com/iorixxx`), so that others could download and compare using the same code used in the present dissertation.

6. EXPERIMENTAL RESULTS AND ANALYSIS

6.1. Introduction

The previous chapter presented the methodology we adopted for the experimental evaluation of the proposed selective term-weighting approach/framework. This chapter presents the experimental evaluation results of the proposed selective term-weighting method.

6.2. Evaluation Criteria

We have evaluated our selective-term weighting approach in three aspects: mean retrieval effectiveness, classification accuracy, and robustness.

6.2.1. Mean retrieval effectiveness

We used two retrieval effectiveness metrics: NDCG@100 and MAP, averaged over the query set. NDCG is a widely used metric in academic research especially when graded levels of relevance labels are available. For example, NDCG is usually preferred to optimize in listwise learning to rank approaches (Valizadegan et al., 2009). MAP is the standard metric for binary relevance judgments.

6.2.2. Classification accuracy

The accuracy of a system is measured by the proportion (%) of queries that the selective-term weighting approach correctly predicted the best model that attained highest effectiveness score. During accuracy evaluation we used two modes: strict mode and relaxed by an one Standard Error (SE). SE given by the Equation 6.1 is equals to the standard deviation (i.e., square root of the variance) divided by the square root of the sample size.

$$SE = \sqrt{\frac{variance}{n}} = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}} \tag{6.1}$$

Table 6.1. Six sample queries: weighting models are sorted by NDCG

ID	SE	First	Second	Third	Fourth	Fifth	Sixth	Seventh	Eighth
173	0.0278	+PL2(0.31775)	+DFIC(0.31767)	BM25(0.27738)	Dirichlet(0.23286)	LGD(0.23152)	DLH13(0.16217)	DPH(0.15722)	-DFRee(0.10403)
129	0.0687	+DPH(0.84343)	DFRee(0.72639)	DLH13(0.61386)	BM25(0.52483)	LGD(0.43028)	Dirichlet(0.3887)	PL2(0.36981)	-DFIC(0.27477)
143	0.0737	+DPH(0.72493)	+DFRee(0.68348)	LGD(0.60148)	DLH13(0.54746)	PL2(0.50847)	Dirichlet(0.47869)	BM25(0.39058)	-DFIC(0.05896)
140	0.0768	+DFRee(0.464)	+DPH(0.46114)	+DLH13(0.38788)	BM25(0.31229)	-Dirichlet(0.04122)	-DFIC(0.0)	-LGD(0.0)	-PL2(0.0)
112	0.0877	+BM25(1.0)	+DFRee(1.0)	+DPH(1.0)	DLH13(0.63093)	Dirichlet(0.63093)	-LGD(0.5)	-PL2(0.5)	-DFIC(0.43068)
19	0.1176	+Dirichlet(0.72836)	+LGD(0.72614)	+DLH13(0.7122)	DFIC(0.48384)	DFRee(0.14187)	PL2(0.1177)	-BM25(0.0)	-DPH(0.0)

where σ is the sample standard deviation and n is the size (number of term-weighting models) of the sample. Table is the query × model table, which presents NDCG@100 values and their associated SE values. Term-weighting models are sorted by their NDCG@100 scores, which are shown inside the parenthesis. Best model(s) that lie within one SE range are marked with plus (+) symbol. These are the best models according to the relaxed mode. For example, relaxed mode considers PL2 and DFIC as the most effective for the query 173. By contrast, the most effective is the PL2 according to the strict mode. Strict mode accepts only one best model (winner) for a query unless there is a tie. In case of a tie, as in the query 112, there can also be multiple winners for the strict mode. Note that relaxed mode implicitly handles ties.

Although we report both modes, we use the relaxed mode for the evaluation of the selective approach, since the models that are in the range "within one standard error of the maximum" can be considered best.

6.2.3. Robustness

An evaluation methodology that focuses on poorly performing (ineffective) topics is needed to support research on "Robustness in Retrieval Performance," which aims at obtaining more consistent retrieval performance across topics (Voorhees, 2004). Using *arithmetic mean* of traditional evaluation measures is not an appropriate methodology for *robustness* because it emphasizes effective topics: poorly performing topics' scores are by definition small, and they are therefore dominated by the effective topics' scores in retrieval evaluation. Reliably measuring the robustness of a system, i.e. its worst-case effectiveness, is important but inherently difficult. To address this, various robust and risk-sensitive measures haven been

proposed. For instance, in the TREC Robust tracks, geometric mean average precision (Robertson, 2006; Ravana and Moffat, 2008) was used to measure the extend to which a system is successful on all queries. More recently, the risk-sensitive evaluation has been introduced, in which robustness of a system is measured by considering per-query losses and wins against a particular given baseline system. In this line of research, U_{Risk} (Wang et al., 2012) and T_{Risk} (Dinçer et al., 2014) measures have been proposed. However, since they only consider a single baseline, they are more appropriate for *before-and-after* experiments. Before is the bare system, which is used as a baseline. After is the system that is obtained by applying some retrieval technique such as stemming, query expansion, diversification, personalization etc.

Dinçer et al. (2016) proposed the new Z_{Risk} robustness measures that takes into account multiple baseline systems when measuring risk, and a derivative measure called *GeoRisk* that enhances Z_{Risk} by taking into account the overall magnitude of effectiveness. In the present dissertation, due to the existence of multiple baseline term-weighting models, GeoRisk is most naturally applicable to measure robustness. For this reason, we use the GeoRisk measure for the robustness criteria.

6.3. Statistical Significance Testing

We conducted two different statistical hypothesis tests of two kinds: parametric and nonparametric. The nonparametric methods require few or no assumptions about the populations from which data are obtained (Hollander and Wolfe, 1999). A paired t-test (Kreyszig, 1970) is representative of parametric testing, while the Wilcoxon signed-rank test (Gibbons and Chakraborti, 2010) is a nonparametric statistical hypothesis test. Both of the tests are employed at a 95% confidence level (p <0.05) to determine statistically significant effectiveness differences. It is important to conduct both parametric and nonparametric significance tests and check whether they agree or disagree on the significance result. If both tests agree on the significance, we can be more confident that the difference of the two systems

Table 6.2. Selective term-weighting result for ClueWeb{09A|12B} dataset (`Anchor`) over 285 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.19559	0	0.31501	0
ℵ†**SEL**	28.42	47.72	1	0.16885	1	0.29082	1
LGD (c=5.0)	13.68	31.58	4	0.15701	2	0.28007	2
BM25 (k_1=1.4 b=0.2)	28.77	43.16	2	0.15662	3	0.27953	3
PL2 (c=12.0)	12.63	34.74	3	0.15543	4	0.27908	4
RMLE	17.16	30.76	5	0.14863	5	0.27238	5
Dirichlet (μ=800)	6.32	20.35	9	0.14697	6	0.27084	6
DPH	19.65	30.18	6	0.14668	7	0.27053	7
RND	13.35	26.37	7	0.14512	8	0.26911	8
DFIC	8.07	22.46	8	0.13647	10	0.26164	9
DFRee	10.88	18.95	10	0.13747	9	0.26149	10
DLH13	6.32	8.77	11	0.12448	11	0.24830	11

that are compared is not caused by the chance fluctuation.

6.4. Query Set Partition Procedure

To split the available query set into the training and test sample we employed the *leave-one-out* procedure/strategy, which is the most classical exhaustive cross-validation procedure (Arlot and Celisse, 2010). Each query is successively "left out" at a time from the query set and used for testing. The training sample is used for training the algorithm, and the remaining test query, which plays the role of yet-unseen data, is used for evaluating the performance of the algorithm. Given only a limited amount of query is available, omitting each query in turn and using the remaining subset for training purposes is a maximal use of the query set at hand because only one query is omitted at each step. Moreover, the procedure is deterministic since no sampling is involved.

6.5. Web Track Results

In this section, we combine ClueWeb09A and ClueWeb12-B13 datasets in order to represent the six Web tracks ran through 2009 to 2014. This resulted in total

Table 6.3. Selective term-weighting result for ClueWeb{09A|12B} dataset (`Anchor`) over 290 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.10559	0	0.23111	0
ℵ†**SEL**	34.83	48.97	1	0.09260	1	0.21518	1
BM25 (k_1=1.2 b=0.3)	31.72	43.10	2	0.08573	2	0.20685	2
PL2 (c=8.0)	15.52	34.83	3	0.08205	3	0.20295	3
LGD (c=3.0)	9.66	24.48	6	0.08183	4	0.20241	4
RMLE	18.55	29.67	4	0.07890	6	0.19854	5
DPH	16.90	24.14	7	0.07902	5	0.19819	6
RND	13.28	23.32	8	0.07576	7	0.19449	7
Dirichlet (μ=500)	4.14	14.14	9	0.07464	8	0.19318	8
DFRee	9.66	14.14	10	0.07337	9	0.19059	9
DFIC	14.48	25.17	5	0.06810	10	0.18535	10
DLH13	3.79	5.86	11	0.06191	11	0.17515	11

Table 6.4. Selective term-weighting result for Million Query 2009 dataset (`Anchor`) over 528 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.44984	0	0.48053	0
ℵ†**SEL**	26.70	40.53	1	0.37157	1	0.43123	1
ℵDPH	29.36	40.34	2	0.35848	2	0.42552	2
DFRee	16.67	29.92	5	0.35795	3	0.42319	3
BM25 (k_1=1.6 b=0.5)	27.46	38.45	3	0.35718	4	0.42216	4
RMLE	19.80	30.55	4	0.34451	5	0.41495	5
LGD (c=1.0)	6.63	17.42	9	0.33958	6	0.41126	6
RND	14.19	24.52	6	0.33403	7	0.40798	7
DLH13	10.04	20.27	8	0.32922	8	0.40360	8
PL2 (c=8.0)	11.55	20.64	7	0.32339	9	0.40036	9
Dirichlet (μ=200)	4.55	14.58	10	0.32166	10	0.39945	10
DFIC	7.01	13.83	11	0.28339	11	0.37505	11

Table 6.5. Selective term-weighting result for Million Query 2009 dataset (`Anchor`) over 542 queries. Retrieval effectiveness is measured by statMAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	statMAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.31299	0	0.39840	0
ℵ†**SEL**	30.26	39.67	1	0.23770	1	0.34481	1
ℵ†DPH	22.88	32.29	3	0.22954	2	0.33863	2
†DFRee	18.08	26.01	5	0.22890	3	0.33806	3
BM25 (k_1=1.6 b=0.4)	30.63	38.93	2	0.22531	4	0.33583	4
RMLE	19.30	26.79	4	0.21563	5	0.32820	5
LGD (c=2.0)	9.96	16.79	8	0.20587	6	0.32091	6
RND	13.88	20.31	6	0.20343	7	0.31861	7
PL2 (c=10.0)	11.99	19.00	7	0.19604	8	0.31272	8
DLH13	9.23	12.92	9	0.19469	9	0.31154	9
Dirichlet (μ=500)	5.17	9.23	10	0.18918	10	0.30685	10
DFIC	3.51	7.38	11	0.15884	11	0.28107	11

Table 6.6. Selective term-weighting result for ClueWeb09A dataset (`Anchor`) over 194 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.22693	0	0.34006	0
ℵ†**SEL**	23.71	43.81	1	0.19449	1	0.31206	1
LGD (c=5.0)	11.34	34.02	4	0.18081	2	0.30062	2
PL2 (c=14.0)	15.46	38.66	2	0.17806	3	0.29884	3
BM25 (k_1=1.0 b=0.3)	20.10	35.57	3	0.17739	4	0.29711	4
RMLE	15.47	29.24	6	0.16873	5	0.29008	5
Dirichlet (μ=800)	6.19	21.13	10	0.16800	6	0.28958	6
DPH	22.16	29.90	5	0.16765	7	0.28894	7
RND	13.26	26.98	7	0.16664	8	0.28831	8
DFIC	9.28	23.71	8	0.15652	10	0.28039	9
DFRee	13.40	21.65	9	0.15767	9	0.27985	10
DLH13	7.73	11.34	11	0.14599	11	0.26889	11

Table 6.7. Selective term-weighting result for ClueWeb09A dataset (`Anchor`) over 197 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.13422	0	0.26089	0
ℵ† **SEL**	34.52	50.76	1	0.11852	1	0.24351	1
BM25 (k_1=1.0 b=0.3)	31.98	43.65	2	0.10962	2	0.23389	2
LGD (c=3.0)	9.64	27.92	6	0.10366	3	0.22797	3
PL2 (c=5.0)	6.60	29.44	4	0.10084	4	0.22474	4
RMLE	19.32	30.52	3	0.09909	6	0.22245	5
DPH	16.24	25.89	7	0.09976	5	0.22225	6
RND	12.95	24.61	8	0.09608	7	0.21899	7
Dirichlet (μ=500)	2.54	14.72	10	0.09445	8	0.21739	8
DFRee	11.17	17.26	9	0.09324	9	0.21452	9
DFIC	19.80	28.93	5	0.08613	10	0.20887	10
DLH13	4.57	7.61	11	0.07981	11	0.19893	11

Table 6.8. Selective term-weighting result for ClueWeb09B dataset (`Anchor`) over 192 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.23860	0	0.34793	0
ℵ† **SEL**	23.96	44.27	1	0.20157	1	0.31710	1
BM25 (k_1=1.6 b=0.3)	23.44	41.67	2	0.19033	2	0.30781	2
LGD (c=5.0)	8.85	28.13	5	0.18679	3	0.30553	3
PL2 (c=30.0)	9.38	25.52	7	0.17990	4	0.29982	4
DPH	27.60	34.38	3	0.17843	6	0.29916	5
RMLE	17.52	30.40	4	0.17887	5	0.29888	6
RND	12.80	26.38	6	0.17522	7	0.29577	7
Dirichlet (μ=800)	7.81	19.79	9	0.17250	8	0.29327	8
DFRee	13.02	23.96	8	0.17189	9	0.29288	9
DLH13	7.29	17.19	11	0.16215	10	0.28421	10
DFIC	4.17	19.79	10	0.16069	11	0.28328	11

Table 6.9. Selective term-weighting result for ClueWeb09B dataset (`Anchor`) over 192 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.13046	0	0.25663	0
ℵ† **SEL**	41.67	52.60	1	0.11338	1	0.23800	1
†BM25 (k_1=1.8 b=0.2)	40.63	51.04	2	0.10765	2	0.23182	2
LGD (c=5.0)	9.38	26.04	6	0.10119	3	0.22499	3
RMLE	23.89	34.08	3	0.09841	4	0.22169	4
PL2 (c=12.0)	5.73	27.08	5	0.09548	5	0.21834	5
DPH	21.88	28.65	4	0.09397	6	0.21667	6
RND	12.77	24.05	7	0.09261	7	0.21508	7
Dirichlet (μ=500)	4.69	16.15	10	0.09045	8	0.21249	8
DFRee	11.46	17.19	8	0.08988	9	0.21166	9
DFIC	4.69	17.19	9	0.08290	10	0.20374	10
DLH13	4.17	9.90	11	0.08099	11	0.20106	11

Table 6.10. Selective term-weighting result for ClueWeb12-B13 dataset (`Anchor`) over 91 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.12660	0	0.25252	0
†PL2 (c=10.0)	9.89	30.77	6	0.10856	1	0.23310	1
ℵ† **SEL**	35.16	47.25	1	0.10817	2	0.23286	2
ℵ†LGD (c=8.0)	13.19	32.97	3	0.10674	3	0.23089	3
†Dirichlet (μ=2000)	16.48	32.97	4	0.10553	4	0.22985	4
†BM25 (k_1=1.6 b=0.2)	35.16	42.86	2	0.10451	5	0.22858	5
RMLE	19.95	32.85	5	0.10301	6	0.22692	6
†DPH	16.48	29.67	7	0.10197	7	0.22578	7
RND	13.40	25.81	8	0.09922	8	0.22260	8
DFRee	6.59	16.48	9	0.09441	9	0.21684	9
DFIC	6.59	15.38	10	0.09374	10	0.21639	10
DLH13	2.20	4.40	11	0.07862	11	0.19761	11

Table 6.11. Selective term-weighting result for ClueWeb12-B13 dataset (`Anchor`) over 93 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.04461	0	0.14997	0
ℵ†PL2 (c=8.0)	10.75	23.66	6	0.03655	1	0.13521	1
ℵ†LGD (c=5.0)	19.35	24.73	5	0.03634	2	0.13473	2
†BM25 (k_1=1.2 b=0.2)	38.71	47.31	2	0.03547	3	0.13320	3
ℵ†**SEL**	37.63	48.39	1	0.03540	4	0.13310	4
ℵ†Dirichlet (μ=1500)	11.83	25.81	4	0.03529	5	0.13290	5
ℵ†DPH	15.05	22.58	7	0.03509	6	0.13244	6
ℵ†RMLE	22.05	30.45	3	0.03486	7	0.13200	7
ℵ†RND	13.80	21.19	8	0.03303	8	0.12843	8
DFRee	4.30	9.68	10	0.03130	9	0.12486	9
DFIC	8.60	12.90	9	0.02990	10	0.12221	10
DLH13	2.15	3.23	11	0.02399	11	0.10919	11

297 queries, 12 of which are yielded zero score of NDCG@100 for all eight term-weighting models. There is no clear winner for these topics therefore they are not useful for selective term-weighting experiments. Table 6.2 presents the results for the remaining 285 queries.

`Oracle` represents an oracle experiment in which an oracle selects the most effective model for the query. Thus, the oracle experiment provides an upper-bound on selective term weighting effectiveness.

`SEL` represents our selective approach, which selects the most effective models with the accuracy of 48%.

Naïve Random, `RND`, which is shown at the ninth row, selects a model at random among eight models. Theoretical classification accuracy of such random selection would be $\frac{1}{8} = 12.5\%$. Its accuracy is reported as 13.30% for $\sigma = 0$ on the Table 6.2. The accuracy is slightly greater than 12.5% because some queries have more than one best model due to a tie in the highest NDCG score. Random selection is the most obvious and natural baseline that any selective retrieval approach must beat at the bare minimum.

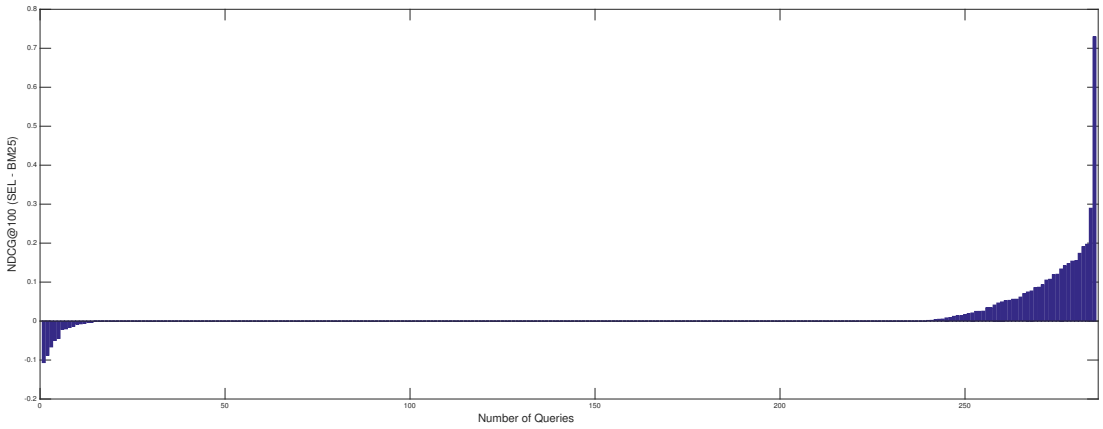

Figure 6.1. ClueWeb{09A|12B} (`Anchor`): Selective term-weighting `SEL` is compared with the `BM25` term-weighting, which is applied uniformly to all 285 queries, in terms of their NDCG@100 differences. Right side of the figure shows the queries that `SEL` performed better than `BM25`.

Second baseline would be the case where randomized selection is performed in a way that takes into account the training data. Random selection based on Maximum Likelihood Estimate `RMLE`, which is shown at the sixth row, selects a model at random by means of favoring the models that were the most effective at the training data. Its accuracy is reported as 17.11% and 30.84% for σ = 0 and σ = 1 respectively on the Table 6.2.

As expected, `RMLE` attains better results than the naïve `RND`. Both randomized selection algorithms are executed/repeated for 1000 times and then average metrics are reported.

The best single term-weighting model, which is applied uniformly to all queries, is the third baseline. However, the best single model can be of two kinds: according to the accuracy and according to the mean retrieval effectiveness. They are the `BM25` model with the accuracy of 43.16% and the `LGD` model with mean effectiveness of 0.15701 on Table 6.2. Our selective term-weighting approach `SEL` brings improvements on both effectiveness (0.16885 - 0.15662 = 0.01223) and accuracy (47.72% - 43.16% = 4.56%) over the `BM25` model. Figure 6.1 is the *risk graph* that shows the difference (between SEL and BM25) in NDCG@100 for 285

queries in the ClueWeb{09A|12B} query set. The query set is sorted by the difference in NDCG@100 magnitudes, which is depicted in the vertical axis. The queries for which the selective approach performed better than BM25 are shown at the right hand side. The queries for which the selective approach performed worse than BM25 are shown at the left hand side. Middle of the figure shows the queries in which both selective and BM25 term-weighting attained the same effectiveness score (e.g., difference is zero).

The improvement that the selective term-weighting approach brings over the BM25 and LGD models is *statistically significant* according to both the t-test and the Wilcoxon signed rank test. The Wilcoxon signed rank test ignores the magnitudes shown in bars of Figure 6.1, instead it counts the number of queries that caused the (negative or positive) difference. For example, the right most query (caused the largest gain over BM25) and the 250^{th} query contribute equally to the Wilcoxon signed rank test. By contrast, the t-test considers the magnitudes, in which contribution of the aforementioned two queries are proportional to their magnitudes.

The selective term-weighting approach performed significantly better than the BM25 model for the sample of 285 queries: p = 0.0003743 for the t-test, p = 0.00000538 for the Wilcoxon signed rank test.

A population can be thought of a complete set of queries (usually infinite in size and unknown in distribution) whereas sample is the subset (i.e., 285 queries in the present experiment) of the query population. The reported p-value is defined as the probability of re-observing a difference in the mean retrieval effectiveness of two systems on a new query sample chosen/drawn from the query population, that is equal to or "more extreme" than what was actually observed on the sample in use, when the two systems have equal population means. Thus, low value of p-value implies statistical significance: the observed difference is not encountered by chance. Usually an upper threshold α of 5% (0.05) or 1% (0.01) is used to compere

with the p-value. It is worthwhile to note that both hypothesis tests (p = 0.0003743 and p = 0.00000538) reject the null the hypothesis (H_0) which claims that BM25 and SEL have the same retrieval effectiveness mean over the population.

In case of the best single term-weighting, which is the `LGD` model, that attained the highest mean effectiveness, our selective term-weighting approach `SEL` brings improvements on both effectiveness (0.16885 - 0.15701 = 0.01184) and accuracy (47.72% - 31.58% = 16.14%) over the `LGD` model. The improvements are statistically significant according to: p = 0.00009015 for the t-test, p = 0.00000433 for the Wilcoxon signed rank test. Figure 6.2 is the *risk graph* between SEL and LGD for 285 queries in the ClueWeb{09A|12B} query set.

Table 6.2 is already sorted the by GeoRisk measure that quantifies retrieval robustness. As expected, the hypothetical oracle run is at the first seat. Other than oracle, our selective term-weighting is the most robust system. Moreover, the selective term-weighting is statistically different from the *all* models listed in Table 6.2 according to both the t-test and the Wilcoxon signed rank test. Thus, the selective term-weighting is the most accurate, effective and robust among all models.

Note that we don't report p-values of all hypothesis tests that we conducted for the sake of the flow of the dissertation. Instead we mark the models in which hypothesis tests fail to reject the null hypothesis. We use † symbol for the paired t-test and א symbol for the Wilcoxon signed-rank test. The absence of these symbols on a table means that the selective approach is statistically different from all the models. If a model is marked with both symbols, it means that it would have the same retrieval effectiveness mean over the population with the selective approach (i.e., there is no significant difference between them).

Table 6.3 presents selective term-weighting experiments where MAP is used as the target effectiveness measure that is optimized. We don't repeat risk graphs and p-values of individual hypothesis tests. Inline with the NDCG@100 results, our

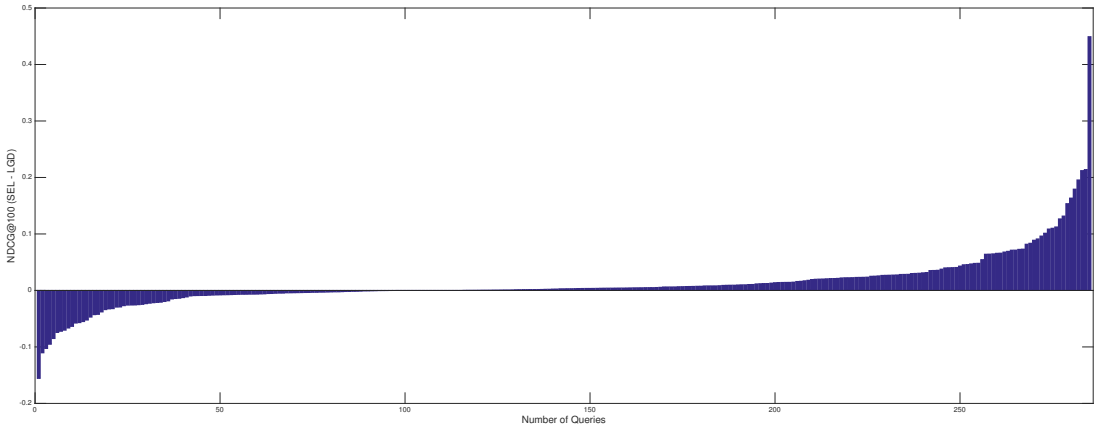

Figure 6.2. ClueWeb{09A|12B} (`Anchor`): Selective term-weighting `SEL` is compared with the `LGD` term-weighting, which is applied uniformly to all 285 queries, in terms of their NDCG@100 differences. Right side of the figure shows the queries that `SEL` performed better than `LGD`.

selective approach was significantly better than the all models when effectiveness is measured by the MAP metric.

6.6. Million Query Results

In this section, we present the results of the Million Query 2009 track, which has 561 valid queries. Again, we discard the queries that the all eight term-weighting models attained the same effectiveness. Table 6.4 presents the results where effectiveness was measured by NDCG@100. Although our selective approach attained the highest scores for all three criteria (accuracy, mean effectiveness, robustness), hypothesis tests didn't agree on the statistical difference from the `DPH` model only. According to the t-test (p = 0.02468486) the difference is significant while according to the Wilcoxon signed rank test (p = 0.07326688) it is not. It is worthwhile to present and examine the per-query risk graph (Figure 6.3) of SEL and DPH in order to understand the principle difference between Wilcoxon and Student's t-test and why/when they do not agree. The graph is almost symmetric about the y-axis: positive gains (wins) are roughly equal to the negative losses. However, as can be seen on the right-most of the graph, selective approach caused large gains (in magnitude) for a few queries. The mean effectiveness difference between SEL

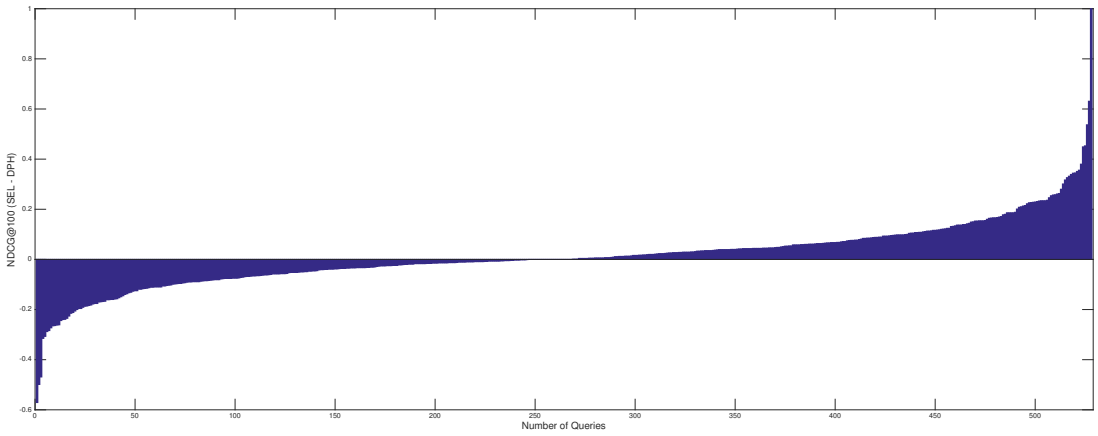

Figure 6.3. Million Query 2009 (`Anchor`): Selective term-weighting `SEL` is compared with the best single term-weighting `DPH` in terms of their NDCG@100 differences. Right side of the figure shows the queries that the selective approach performed better than `DPH`.

and DPH (0.37157 - 0.35848 = 0.01309) is mostly caused by these queries. Since t-test considers magnitudes, the difference was significant according to it. Wilcoxon rather considers the number of hurt and improved queries, and the difference was not significant according to it. Thus, Wilcoxon considers retrieval robustness to some extend. This formed a perfect example to demonstrate that Wilcoxon and Student's t-test are different in nature.

Table 6.5 presents the Million Query result in which effectiveness is measured by the statMAP measure. Inline with the NDCG@100 results, SEL approach was not statistically different than DFRee (t-test's p_t=0.18737933 Wilcoxon's p_w=0.00483313) and DPH (t-test's p_t=0.15644347 Wilcoxon's p_w=0.24048373). The per-query risk graph (Figure 6.4) of SEL and DFRee demonstrates the other edge case: significant according to Wilcoxon, but not significant according to Student's t-test. This graph is symmetric at the right and left edges. However, at the middle range, SEL approach caused gains (small in magnitude) for the queries roughly between 350 and 400. The disagreement was primarily influenced by these 50 queries.

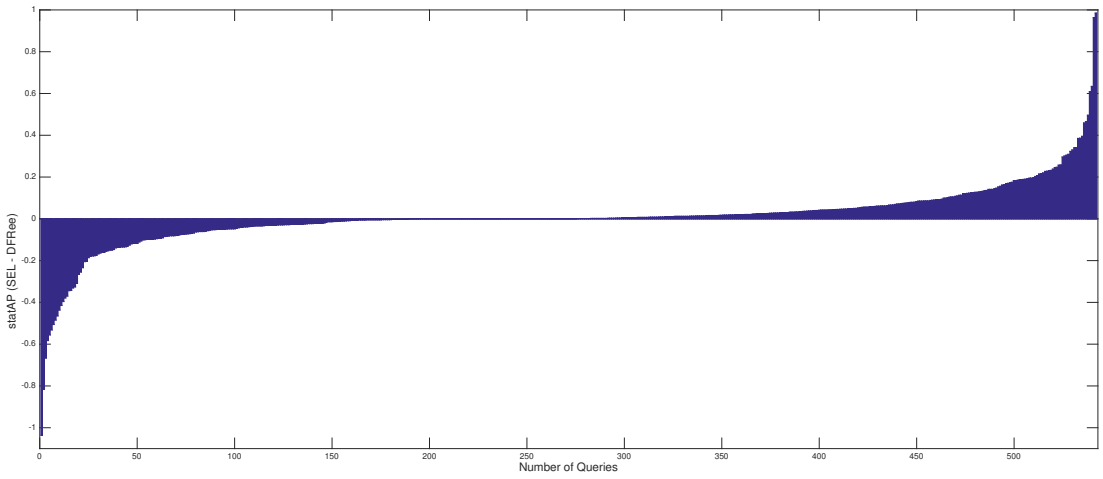

Figure 6.4. Million Query 2009 (`Anchor`): Selective term-weighting `SEL` is compared with the `DFRee` model in terms of their statAP differences. Right side of the figure shows the queries that the selective approach performed better than `DFRee`.

6.7. Within-Collection Experiments Results

In this section we present results of selective term-weighting experiments conducted on individual datasets.

6.7.1. ClueWeb09-A

The results of the ClueWeb09 full dataset are presented on Table 6.6 and Table 6.7 for the NDCG@100 and MAP respectively. Again the SEL approach attains the highest scores for all three criteria (accuracy mean effectiveness, robustness), and significantly different from all the models. Thus, SEL performed significantly better than all the models for both NDCG@100 and MAP measures. Since we did not encounter a disagreement between two hypothesis tests, we do not give p-values nor risk graphs.

6.7.2. ClueWeb09-B

The results of the category B subset of the ClueWeb09 dataset are presented on Table 6.8 and Table 6.9 for the NDCG@100 and MAP respectively. Again the SEL approach attains the highest scores for all three criteria (accuracy mean effective-

ness, robustness). However for the MAP metric, hypothesis tests disagree on the BM25 model (p_{t-test}=0.10958176 and $p_{wilcoxon}$=0.03362139). Thus, SEL performed significantly better than all the models for NDCG@100; while its MAP difference from BM25 was not statistically significant according to the Student's t-test.

6.7.3. ClueWeb12-B13

The results of the category B subset of the ClueWeb12 dataset are presented on Table 6.10 and Table 6.11 for the NDCG@100 and MAP respectively. For this dataset the SEL approach is not significantly different than most of the models (BM25, DPH, Dirichlet, LGD, PL2). There can be two reasons for this: (i) The number of queries ($\approx$ 90) are too few (ii) mean effectiveness scores of the models are small in magnitude due to the category B subset. On the other hand, SEL approach took the first seat in the accuracy rank.

6.7.4. Summary

In this section we sum-up within-collection experiments conducted for NDCG and MAP measures. Classification accuracy (σ = 1) attained in each collection and in each effectiveness measure are depicted on Figure 6.5. The interaction plot includes three models: our selective approach (SEL), the best single model (SGL) and the MLE random selection (RMLE). It can be clearly seen that our selection mechanism is far away from being random. Moreover, our selective term-weighting (SEL) systematically outperforms the best single term-weighting (SGL) for all datasets and measures. Recall that, the differences are statistically significant for most of the datasets and measures.

6.8. Cross-Collection Experiments Results

In this section we investigate the transferability of term frequency distribution data extracted from one corpus to other corpora or subsets of the same corpus.

For instance, such transferability issue has arisen during the TREC Web track 2013 when the new ClueWeb12 dataset was introduced for the first time.

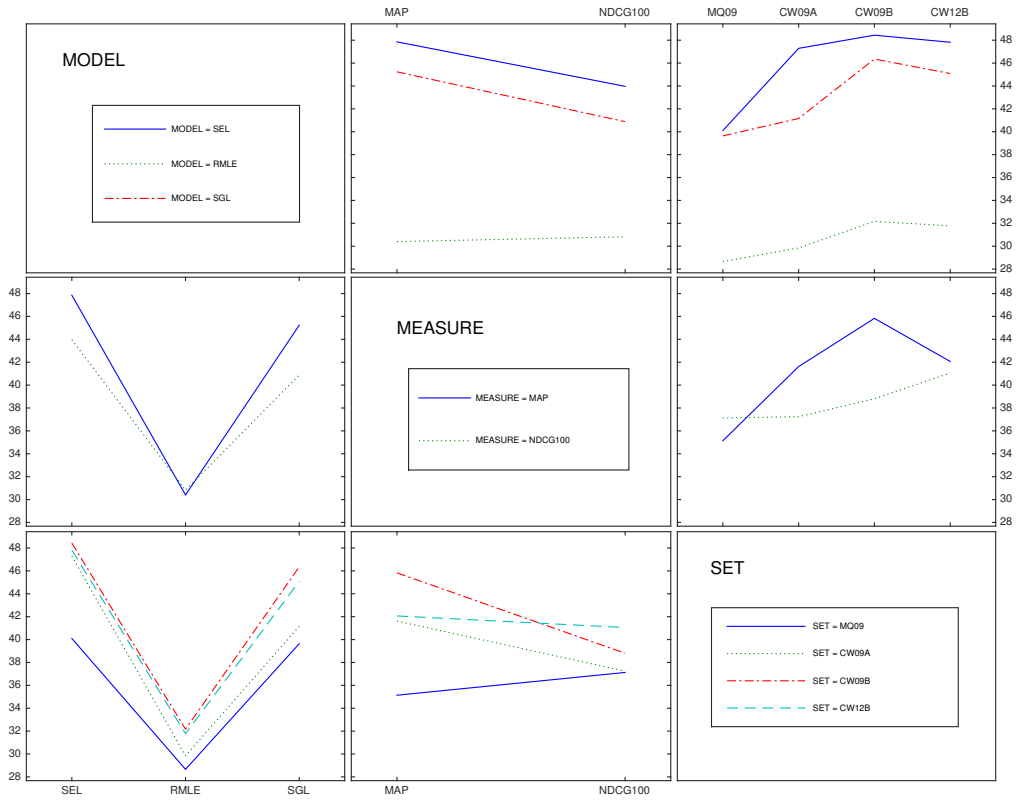

Figure 6.5. Classification Accuracy Interaction Plot of within-collection experiments: `SEL` represents the selective term-weighting, `SGL` represents the best single term-weighting, `RMLE` is the random selection based on MLE.

Since there were no training data available for the ClueWeb12 dataset, participants had to use features extracted from the ClueWeb09 dataset for training purposes. Arguello et al. (2016) stated that "the science of determining similarity between collections and therefore predicting which system components will work well on a new collection is in its infancy."

To empirically investigate these questions we apply a full-factorial experimental design whose factors and levels are given on Table 6.12. We assume that we don't have any relevance information about the test dataset: we use relevance judgments of training data (which model is the most effective at which query). The frequency distributions of the test queries are extracted from the test dataset while the frequency distributions of the training queries are extracted from the training

Table 6.12. Factors and corresponding levels considered in cross-collection experiments.

Factor	Level	Code
Train	Million Query 2009	MQ09
	ClueWeb09 Category A	CW09A
	ClueWeb09 Category B	CW09B
	ClueWeb12 Category B	CW12B
Test	Million Query 2009	MQ09
	ClueWeb09 Category A	CW09A
	ClueWeb09 Category B	CW09B
	ClueWeb12 Category B	CW12B
Effectiveness	Mean Average Precision	MAP
	Discounted Cumulative Gain	NDCG

dataset.

Classification accuracy (σ = 1) attained by the SEL model in each collection and in each effectiveness measure are depicted on Figure 6.6. Empirical results suggest that ClueWeb09 full dataset forms the best candidate for training. ClueWeb12-B13 dataset is also a good candidate. Given that ClueWeb12-B13 is random sample of the full dataset, and ClueWeb09A is full dataset itself, frequency distributions extracted from uncontrolled datasets are more useful for selective term-weighting based on term frequency distributions. Recall that category B subset of the ClueWeb09 is selected in a deterministic way (i,e first 50 million pages plus entire English Wikipedia), therefore CW09B is not as representative as CW12B of the full datasets. Once you buy the ClueWeb09 full dataset, you can deterministically extract the category B subset. You cannot do the same with the ClueWeb12 full dataset.

6.9. The Role of Anchor Text

Anchor Text is the visible, clickable text in a hyperlink that can be tough of the edges in a graph whose nodes are the Web pages. The HTML excerpt in Listing 6.1 is an example of an anchor text, in which `href` attribute specifies the URL of the page the link goes to. Once the HTML is rendered, say by a Web browser, one can see only the clickable anchor text: "*Visit our HTML tutorial*."

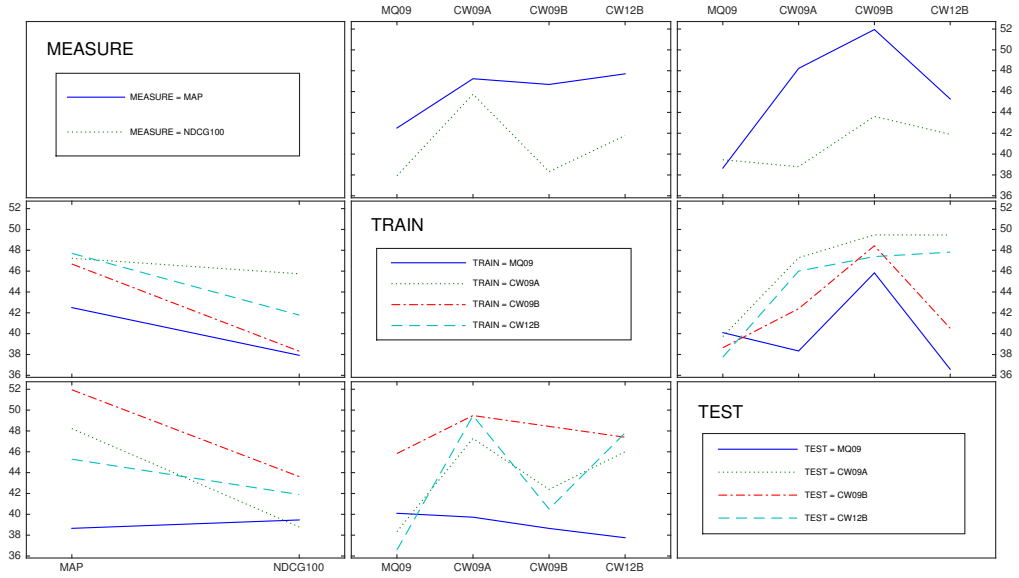

Figure 6.6. Classification Accuracy Interaction Plot of cross-collection experiments.

Listing 6.1. Example of an Anchor Text

```
<a href="http://www.w3schools.com/html">Visit our HTML tutorial</a>
```

Anh and Moffat (2010) investigated the role of Anchor Text in ClueWeb09 retrieval and found that the use of an anchor text significantly increase the retrieval effectiveness.

In previous sections, as in common practice, we index anchor texts from incoming links as parts of the documents. So one point of difference from the body of the document is that it is not written by the author of the document, but by the author of the source. Another point is that anchor text forms a *repeatable field*, which means there can be any number (zero or more) of incoming links (anchor texts) for a given document (Robertson et al., 2004).

In general, the number of incoming links are considered as the indicator of quality/authority of the page. Robertson et al. (2004) warned against the *swamp* effect: large number of anchor texts may swamp the remainder of the document.

In this section, we investigate the role of anchor text on term frequency distributions and therefore selective term-weighting. We present selective term-

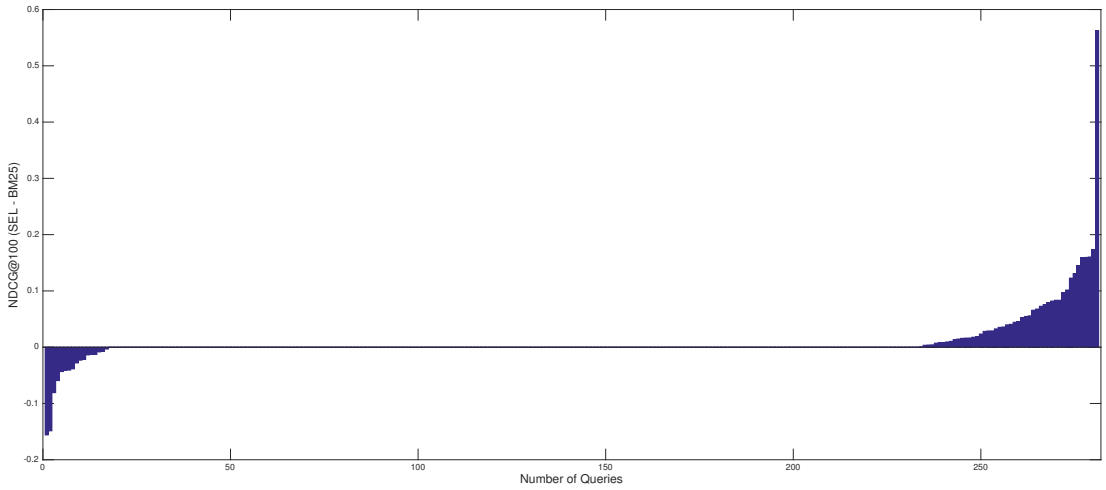

Figure 6.7. ClueWeb{09A|12B} (`NoAnchor`): Selective term-weighting `SEL` is compared with the `BM25` term-weighting, which is applied uniformly to all 285 queries, in terms of their NDCG@100 differences. Right side of the figure shows the queries that `SEL` performed better than `BM25`.

weighting results when anchor text is not used.

6.9.1. Web track results

The results of the six Web tracks ran through 2009 to 2014 are presented on Table 1 and Table 2 for the NDCG@100 and MAP respectively. Figure 6.7 is the *risk graph* between SEL and BM25 for 281 queries for NDCG@100 measure. Although SEL is statistically better than the all models for the NDCG@100, it is not statistically different from PL2 (p_t=0.32297339 p_w=0.79850608) for the MAP metric.

6.9.2. Million query results

The results of the Million Query 2009 track are presented on Table 3 and Table 4 for the NDCG@100 and statMAP respectively. Figure 6.8 is the *risk graph* between SEL and BM25 for 522 queries for NDCG@100 measure. Although SEL is statistically better than the all models for the NDCG@100, it is not statistically different from DFRee, DPH, LGD and PL2 for the MAP metric.

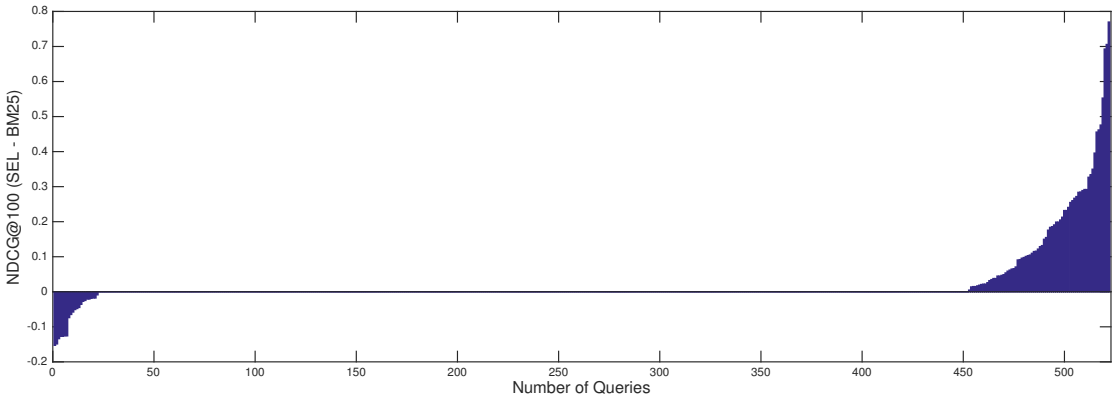

Figure 6.8. Million Query 2009 (`NoAnchor`): Selective term-weighting `SEL` is compared with the `BM25` term-weighting, which is applied uniformly to all 522 queries, in terms of their NDCG@100 differences. Right side of the figure shows the queries that `SEL` performed better than `BM25`.

6.9.3. ClueWeb09-A results

The results of the four Web tracks ran through 2009 to 2012 are presented on Table 5 and Table 6 for the NDCG@100 and MAP respectively. The best performing models are DFIC and PL2 for NDCG@100 and MAP measures respectively. But they are not statistically different than the selective-term weighting: DFIC (p_t=0.27964832 p_w=0.58921883), PL2 (p_t=0.48017690 p_w=0.92429297).

6.9.4. ClueWeb09-B results

The results of the four Web tracks ran through 2009 to 2012 are presented on Table 7 and Table 8 for the NDCG@100 and MAP respectively. For this dataset, selective term-weighting failed to create a significant difference from DFIC, PL2, and LGD.

6.9.5. ClueWeb12-B13 results

The results of the two Web tracks ran through 2013 to 2014 are presented on Table 9 and Table 10 for the NDCG@100 and MAP respectively. For this dataset, selective term-weighting failed to create a significant difference from most of the models.

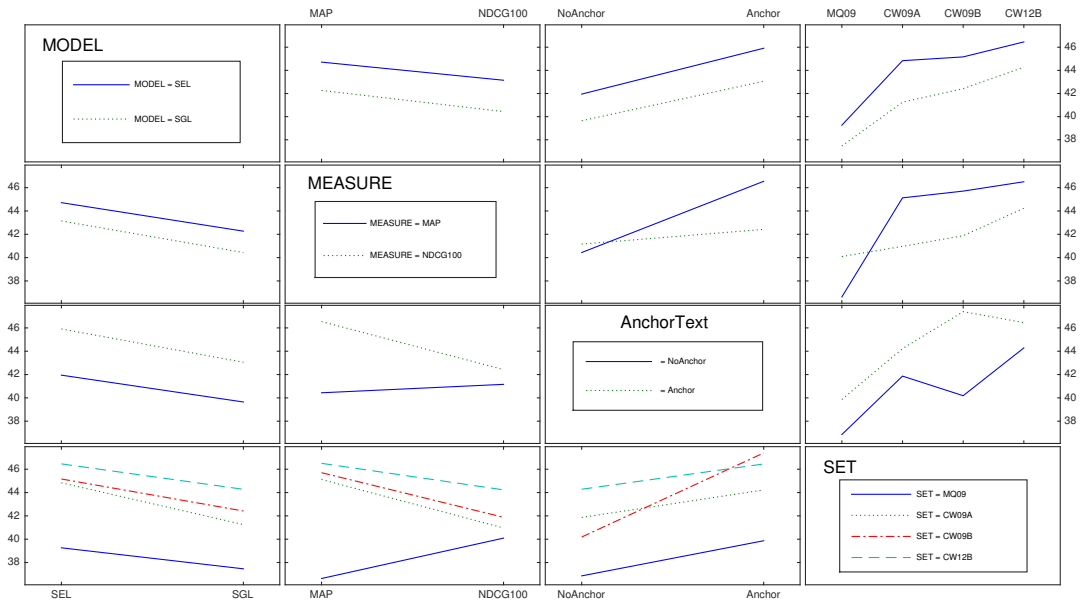

Figure 6.9. Classification Accuracy Interaction Plot of within-collection experiments: `SEL` represents the selective term-weighting and `SGL` represents the best single term-weighting.

6.9.6. Summary

Consistently, when anchor text is not used, selective term-weighting failed to create a significant difference from some of the single term-weighting models that uniformly applied to all queries. Experimental results clearly show that selective approach benefits from the anchor text. Anchor text 100% supports the fundamental assumption of the frequentist approach to IR: the more occurrence of a word implies the more document treats the subject. Besides term frequency, every increase in frequency of a term in anchor text means an additional incoming link, which means some other page in the Web endorsed the target page with the same keyword. Thus, term frequency in anchor text also qualifies the document quality.

The benefit gained from the inclusion of the anchor text can be observed more clearly on Figure 6.9 and Figure 6.10 for within-collection experiments and cross-collection experiments respectively.

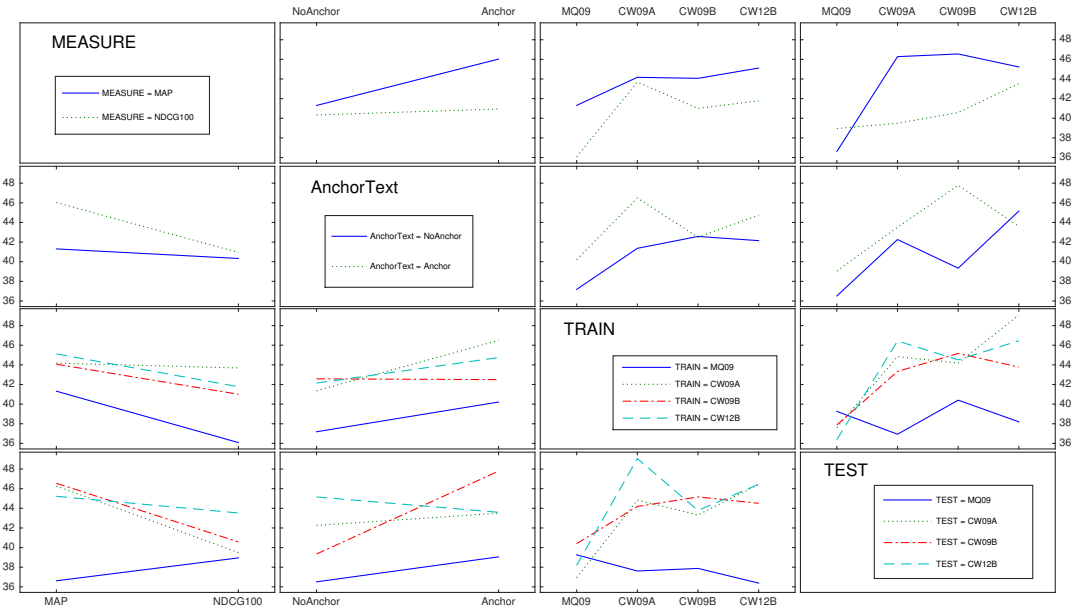

Figure 6.10. Classification Accuracy Interaction Plot of cross-collection experiments for selective term-weighting approach `SEL`.

6.9.7. Overall evaluation

To summarize the overall performance of the proposed selective term-weighting (SEL) approach in Accuracy, Effectiveness, Robustness (AER) criteria at once, we tabulate the ranks of the approach on Table 6.13. Each table entry contains a triple (*rank*@A *rank*@E *rank*@R) that represents the ranks at AER.

It can be clearly seen that the SEL approach is at first seat most of the time and is "always" at first seat for the "accuracy" criterion. SEL performed better on NDCG@100 measure and when the Anchor Text is included in the document representation. Indeed, NDCG is a more appropriate measure for the Web search case in which there exists "six shades of relevance." Anchor Text is specific to Web retrieval by representing the number of incoming links. Moreover, for a web page, the content of its incoming anchor text may be better than the content of itself in objectively describing the page.

Among the datasets, SEL performed worst on ClueWeb12-B13 dataset. This is due to too few queries (≈ 90) and too few relevant documents and small effec-

Table 6.13. Overall Summary Table: shows the ranks of selective term-weighting at three criteria: accuracy, effectiveness, robustness.

		CW{09A|12B}	MQ09	CW09A	CW09B	CW12B
Anchor	NDCG	1 1 1	1 1 1	1 1 1	1 1 1	1 2 2
	MAP	1 1 1	1 1 1	1 1 1	1 1 1	1 4 4
NoAnchor	NDCG	1 1 1	1 1 1	1 2 2	1 2 1	1 2 2
	MAP	1 2 2	1 1 1	1 2 2	1 3 3	1 1 1

tiveness measures (we used category A relevance judgments to evaluate category B runs).

To summarize the results the hypothesis tests conducted to decide whether observed mean retrieval effectiveness (*rank*@E) differences are statistically significant, we tabulate the models that are *not* statistically different from the selective approach on Table 6.14. † symbol represents the paired t-test while ℵ symbol represents the Wilcoxon signed-rank test. Empty cells correspond to the case where SEL approach performed statistically better than all in effectiveness criteria. Note that hypothesis tests are used for mean effectiveness which is one the three criteria. Accuracy criterion can be tought of pure robustness: higher value means less significant failure. Effectiveness criterion is pure retrieval effectiveness.

Wilcoxon can be though as significance test for accuracy, while paired t-test is the tool for detecting significant difference in effectiveness. GeoRisk is blended version of both effectiveness and robustness: higher value implies both robust and effective system. There is no significance test for GeoRisk tough.

If we interpret the Table 6.14 on this perspective, for example, although SEL is always at first seat (1 1 1) on the MQ09 Anchor dataset, its accuracy is not significantly from DPH for NDCG@100. Similarly, although SEL is always at first seat (1 1 1) on the CW09B Anchor dataset, its effectiveness is not significantly different from BM25 for MAP measure.

Summary of the Tables 6.14 and 6.13 reveals the fact that SEL is the least risky system: It never performs significantly worse besides it usually performs significantly better. Indeed, this is the ideal behavior of a truly *robust* system.

Table 6.14. The models that are *not* statistically different from the selective model

		CW{09A|12B}	MQ09	CW09A	CW09B	CW12B
Anchor	NDCG		ˢDPH			†PL2,ˢ†LGD,†Dirichlet,†BM25,†DPH
	MAP		ˢ†DPH,†DFRee		†BM25	ˢ†PL2,ˢ†LGD,†BM25,ˢ†Dirichlet,ˢ†DPH,ˢ†RMLE,ˢ†RND
NoAnchor	NDCG			ˢ†DFIC,†DPH,†BM25,†PL2	†LGD,†PL2,†DFIC	ˢ†PL2,ˢ†Dirichlet,ˢ†LGD,†RMLE,†BM25,ˢ†DPH
	MAP	ˢ†PL2,†DFIC	ˢ†DPH,ˢ†PL2,†DFRee,†LGD	ˢ†PL2,†DFIC	ˢ†PL2,ˢ†LGD,†DFIC,†BM25,†Dirichlet	†PL2,ˢ†LGD,ˢ†Dirichlet,†BM25,†DPH,†RMLE

6.10. Similarity? Dissimilarity? or Both?

In Chapter 4, we based our selection mechanism on both similarity and dissimilarity as given by Equation 6.4. However, selection could be done using either similarity or dissimilarity alone as given by Equations 6.5 and 6.6 respectively. In this section, we compere these three methods: selection based on WIN, ODDS, and LOSS.

Inspired by the Probability Ranking Principle (Robertson, 1997), which suggests ranking the documents by the log-odds ratio of their probabilities of being generated by the relevant class against the non-relevant class, selective term-weighting can be formulated as ranking models $m_i \in M$ by increasing probability of performing the best $P(best \mid m_i, q)$ for a given query q.

$$P(best \mid m, q) \propto_q \frac{P(best \mid m, q)}{P(\overline{best} \mid m, q)} = \frac{P(m \mid best, q)P(best \mid q)}{P(m \mid \overline{best}, q)P(\overline{best} \mid q)} \tag{6.2}$$

$$P(best \mid m, q) \propto_q \frac{P(m \mid best, q)}{P(m \mid \overline{best}, q)} = \frac{P(m \mid best, q)}{P(m \mid worst, q)} \tag{6.3}$$

In Equation 6.2, we simply replace the probability by an *odds-ratio* and perform Bayesian inversions on both numerator and denominator. In Equation 6.3, we drop the second component which is independent of the model, therefore does not affect the ranking of the models. And we replace probability of not performing best $P(\overline{best})$ with probability of performing worst $P(worst)$.

We estimate $P(m_i \mid best, q) = \frac{1}{m_i.\texttt{similarity}}$ by comparing the query q with the queries that the model m_i performed best in the training queries (winner list of m_i). We estimate $P(m_i \mid worst, q) = m_i.\texttt{dissimilarity}$ by comparing the query q with the queries that the model m_i performed worst in the training queries (loser

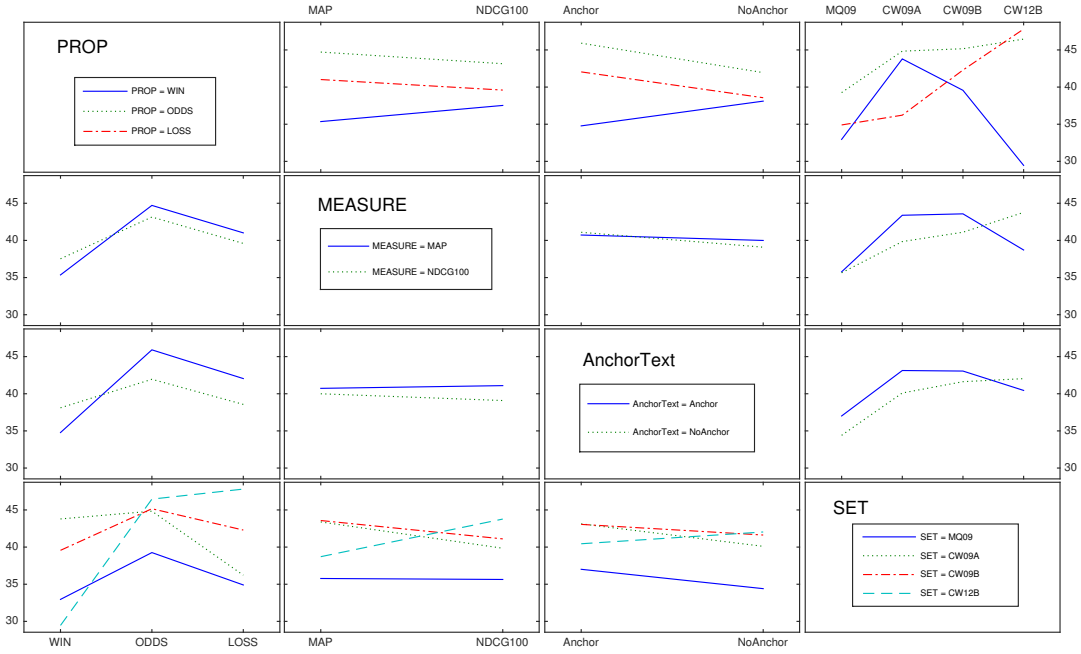

Figure 6.11. Classification Accuracy Interaction Plot of selective term-weighting experiments based on WIN, ODDS, and LOSS.

list of m_i). Thus, $P(best \mid m_i, q) \propto_q \frac{m_i.\texttt{dissimilarity}}{m_i.\texttt{similarity}}$ corresponds to the Equation 6.4 in which we use *odds-ratio*. We name this ODDS. We name Equations 6.5 and 6.6 as WIN and LOSS respectively.

$$\arg\max_{m_i \in M} f(m_i) = \{m_i \mid m_i.\texttt{dissimilarity}/m_i.\texttt{similarity}\} \tag{6.4}$$

$$\arg\min_{m_i \in M} f(m_i) = \{m_i \mid m_i.\texttt{similarity}\} \tag{6.5}$$

$$\arg\max_{m_i \in M} f(m_i) = \{m_i \mid m_i.\texttt{dissimilarity}\} \tag{6.6}$$

Figure 6.11 shows interaction plot for the three probability estimators. It can be seen that ODDS, which uses two sources of information, consistently outperforms to other two. In other words, ranking models by the odds of being the most

effective (best) against the least effective (worst) for a given query optimizes the accuracy of selective term-weighting. This interesting finding suggests that: the queries that a model fails (i.e., performs worst) carry/signal important source of evidence (information) for selective term-weighting based on frequency distribution of query terms.

6.11. Comparison with the Model Selection (MS) Method

In this section we compare our method with the very first selective term-weighting study by He and Ounis (2003, 2004b), whose detailed explanation is given in Section 5.4.2. The model selection method has one free-parameter named k, as in the original work, we use the range of values from 2 to 10. Here k is the number of clusters. Figure 6.12 shows comparison interaction plot including the factor of $k \in \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$. As can be observed from the figure our selective method (SEL) is systematically better than the MS proposed by He and Ounis (2003, 2004b) except for the ClueWeb12-B13 dataset.

To test whether mean retrieval effectiveness differences between SEL and MS (k=7) are statistically significant, we present the results of the hypothesis tests on Table 6.15. In our comparisons, we used $k = 7$ because it is reported as the best threshold setting (He and Ounis, 2004b). In which, $\uparrow$ indicates that mean effectiveness of SEL is greater than MS and $\downarrow$ indicates that mean effectiveness of SEL is less than MS. If the SEL and MS are statistically different ($p < 0.05$) then $\dagger$ symbol (paired t-test) and/or $\aleph$ symbol (Wilcoxon signed-rank test) are inserted. In terms of mean retrieval effectiveness, MS is significantly (Wilcoxon signed-rank test) better than SEL for the ClueWeb12-B13. On the other hand, SEL is significantly (Wilcoxon signed-rank test) better than MS for the ClueWeb09A and ClueWeb09B datasets when anchor text is used. For the Million Query 2009, SEL and MS are not statistically different from each other.

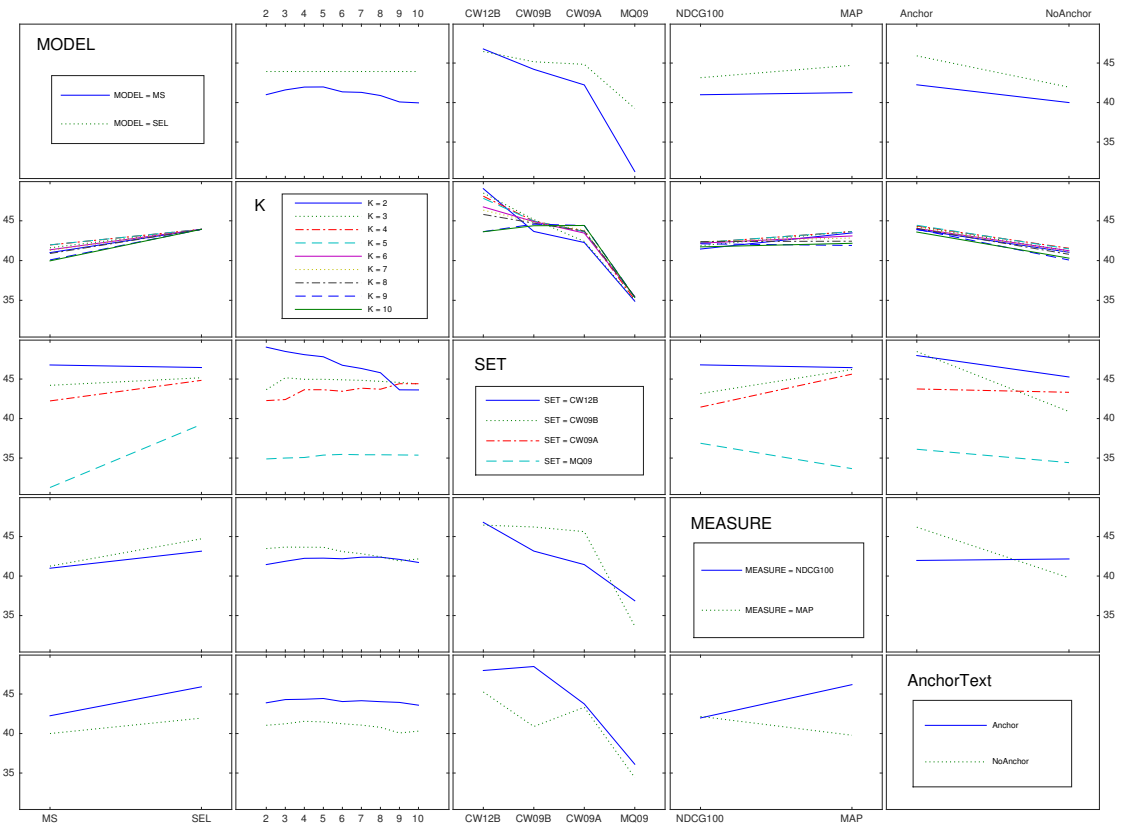

Figure 6.12. Comparison with the Model Selection

6.12. Conclusions

In this chapter, we tested our STW framework on the ClueWeb{09|12} corpora and their corresponding TREC tasks. In particular, we compared it with two random selection mechanisms as well as eight state-of-the-art term-weighting models, namely BM25, DFIC, DLH13, DPH, DFRee, PL2, LGD and the Language Modeling method with Dirichlet prior smoothing. The experimental results showed that,

Table 6.15. The Statistical Significance Tests: MS (k=7) versus SEL

		MQ09	CW09A	CW09B	CW12B
Anchor	NDCG@100	↑	↑א	↑א	↓א
	MAP	↑	↑א†	↑א	↓א
NoAnchor	NDCG@100	↑	↓†	↓א	↓א
	MAP	↓	↑א	↓	↓א

our selective term-weighting approach does improve the average effectiveness compared with a baseline where a single term-weighting model is applied uniformly to all queries.

Our experimental results showed that the retrieval performance obtained by using our proposed STW framework could constantly outperform random selections and eight state-of-the-art models in the NDCG and MAP measures on different datasets. In addition, improvements were statistically significant in most cases.

Moreover, we investigated the robustness of our framework as measured by the GeoRisk. Our experimental results showed that the STW framework is a truly robust system, which avoids significant per-query failures and also maintains a good average effectiveness at the same time. We showed that more robust and more effective system can be built by leveraging existing term-weighting models in a selective manner, without inventing a new one. The STW framework reached to a level of robustness that any single term-weighting cannot posses alone.

Furthermore, we showed that the queries that a system fails is as important as the queries that a system succeeds in a selective IR application. We also empirically validated that the anchor text 100% obeys the fundamental assumption of the frequentist approach to IR.

Finally, we compared our selective term-weighting method with a previous study by He and Ounis (2004b), in which experimental results show that our approach performs better for the ClueWeb09 dataset.

7. CONCLUDING REMARKS

7.1. Contributions and Conclusions

There has been a great deal of research dedicated to develop term-weighting models for information retrieval (IR). However, IR research has shown that there is not a single term-weighting model that would answer the best on any query. Rather, per-query performance fluctuation among the term-weighting models has been shown. This is called robustness problem of IR. This dissertation has investigated the selective application of an appropriate term-weighting model on a per-query basis to alleviate the problem of robustness in retrieval effectiveness. The objective of this dissertation is to characterize queries based on frequency distributions of their terms on document collections and try to predict which term-weighting model would be most effective for each query. Thus, when an unseen query is submitted by the user, our approach will decide which term-weighting model should process it. This section discusses the contributions and conclusions of this dissertation.

7.1.1. Contributions

The main contributions of this dissertation are the introduction of the STW framework and the proposed use of chi-square goodness-of-fit test on frequency distributions of query terms for identifying similar queries. In addition, this dissertation draws insights from a large set of experiments, involving three different standard corpora, two different search tasks, two different document representations and two different effectiveness measures calculated at various cutoff levels. This illustrates the generalizability of the STW framework.

Furthermore, we thoroughly evaluate the accuracy, effectiveness and robustness of the STW framework on two different retrieval tracks, namely Web Track and Million Query Track. In particular, a Web collection that contains over a half billion English documents and about one thousand queries are used in this

evaluation.

This study makes some important contributions to the body of existing work in both selective IR and robust IR. This dissertation presents experiments of the selective term-weighting for robust retrieval based on frequency distributions of query terms. This is the first examination of the frequency distributions of query terms on document collections in text-based IR. This has not been done before. As a by-product, a new family of query features can be driven from the frequency distribution of query terms for to use in IR research area.

This work presents a unique evaluation methodology for selective retrieval approaches when there exist multiple candidates to choose from. Three aspects of such evaluation: accuracy, effectiveness and robustness are considered at the same time. Two natural baselines that any selective retrieval approach should outperform at the minimum are derived and described.

The present dissertation also reveals the organic connection between the selective IR and the robust IR that focused on avoiding significant failures caused by the poorly-performing queries. This connection has much to do with the true understanding/definition of a significant failure, and an appreciation of it helps to gain insight into the selective retrieval approach. Indeed, significant failure is a vague concept. When does a retrieval system fail significantly? Can an effectiveness score of 0.2 or 0.6 be considered a failure for a particular query?

Whether a system performs poorly or not can only be meaningfully identified when it is *relatively* compared to the *other* systems. For example, a system serves a query with the NDCG score of 0.6 and all the other systems attain NDCG score greater than 0.7. Since the model in question is the least effective, we can call NDCG score of 0.6 a significant failure. On the other hand, a system can be the most effective with an NDCG score of 0.2 when the other systems return zero relevant documents. Obviously NDCG score of 0.2 is not a significant failure in this case.

These examples clearly demonstrate that significant failure must be defined in a relative manner. There must be other systems to compare with. The magnitude of an effectiveness score alone is not enough to define it. This is where selective retrieval approaches come into play. The interesting relationship between the selective retrieval approaches and the problem of robustness in retrieval effectiveness is that the selective approaches are natural solutions to the robustness problem.

7.1.2. Conclusions

This section discusses the achievements and the conclusions of this study. We tested our selective term weighting (STW) framework on the ClueWeb{09|12} corpora and their corresponding TREC tasks. In particular, we compared it with two random selection mechanisms as well as eight state-of-the-art term-weighting models, namely BM25, DFIC, DLH13, DPH, DFRee, PL2, LGD and the Language Modeling method with Dirichlet prior smoothing. The experimental results showed that, our selective term-weighting approach does improve the average effectiveness compared with a baseline where a single term-weighting model is applied uniformly to all queries.

Our experimental results showed that the retrieval performance obtained by using our proposed STW framework could constantly outperform random selections and eight state-of-the-art models in the NDCG and MAP measures on different datasets. In addition, improvements were statistically significant in most cases.

Moreover, we investigated the robustness of our framework as measured by the GeoRisk. Our experimental results showed that the STW framework is a truly robust system, which avoids significant per-query failures and also maintains a good average effectiveness at the same time. We showed that more robust and more effective system can be built by leveraging existing term-weighting models in a selective manner, without inventing a new one. The STW framework reached to a level of robustness that any single term-weighting cannot posses alone.

Furthermore, we showed that the queries that a system fails is as important

as the queries that a system succeeds in a selective IR application. We also empirically validated that the anchor text 100% obeys the fundamental assumption of the frequentist approach to IR. Finally, we compared our selective term-weighting method with a previous study by He and Ounis (2004b), in which experimental results show that our approach performs better for the ClueWeb09 dataset.

7.2. Directions for Future Research

This section discusses several directions for future work related to, or stemming from this dissertation.

Query features based on term frequency distributions The most common term statistics currently used in the IR literature are as follows:

- Document frequency: How many documents contain the term?
- Within-collection term frequency: How many times the term is observed in the entire collection?
- Number of documents: How many documents do exist in the entire collection?
- Number of terms: How many terms do exist in the entire collection?

These statistics has been leveraged in IDF and ICTF to quantify term specificity. IDF is the logarithmically scaled inverse fraction of the documents that contain the word and the total number of documents. ICTF simply uses "the number of terms" instead of the number of documents, thus ICTF is the "# terms" counter part of the IDF. Basically they both are based on the mean of the frequency distribution of the term in the collection. However, there are three more statistics that can describe a frequency distribution of a term. The four central moments used in mathematics and statistics are as follows:

- **Mean** is the first raw moment.

- **Variance** measures how far a set of numbers are spread out from their mean.
- **Skewness** is the measure of the lopsidedness of the distribution.
- **Kurtosis** is the measure of the heaviness of the tail of the distribution.

McDonell, Zobel and Billerbeck (2016) argue that the term frequency distribution is pertinent to informativeness and the most informative terms tend to be those whose within-document frequency has high variance across a document collection. The authors propose use of relative standard deviation as a measure of term specificity.

To the best of our knowledge, unlike the mean and variance (McDonell et al., 2016), skewness and kurtosis have not ben utilized yet in the IR literature. The computation of these features is more costly than IDF and ICTF, but they can obviously be included/used to describe terms in various IR tasks, such as query classification and learning to rank. This new family of features based on term frequency distribution are waiting to be exploited by the IR community.

REFERENCES

Amati, G. (2006). Frequentist and bayesian approach to information retrieval, *Advances in Information Retrieval*. Springer Berlin Heidelberg. volume 3936 of *Lecture Notes in Computer Science*, pp. 13–24.

Amati, G. (2009). Divergence from Randomness Models. Springer US, Boston, MA. pp. 929–932.

Amati, G., Carpineto, C., and Romano, G. (2004). Query difficulty, robustness, and selective application of query expansion, *Advances in Information Retrieval*. Springer Berlin Heidelberg. volume 2997 of *Lecture Notes in Computer Science*, pp. 127–137.

Amati, G., and Van Rijsbergen, C.J. (2002). Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.* 20(4), 357–389.

Anh, V.N., and Moffat, A. (2010). The Role of Anchor Text in ClueWeb09 Retrieval. Technical Report. National Institute of Standards and Technology.

Arguello, J., Crane, M., Diaz, F., Lin, J., and Trotman, A. (2016). Report on the SIGIR 2015 workshop on reproducibility, inexplicability, and generalizability of results (RIGOR). *SIGIR Forum* 49(2), 107–116.

Arguello, J., Diaz, F., Lin, J., and Trotman, A. (2015). SIGIR 2015 workshop on reproducibility, inexplicability, and generalizability of results (RIGOR), *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Santiago, Chile. pp. 1147–1148.

Arlot, S., and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys* 4, 40–79.

Balasubramanian, N., and Allan, J. (2010). Learning to select rankers, *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Geneva, Switzerland. pp. 855–856.

Białecki, A., Muir, R., and Ingersoll, G. (2012). Apache Lucene 4, *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*, Portland, Oregon, USA. pp. 17–24.

Bian, J., Liu, T.Y., Qin, T., and Zha, H. (2010). Ranking with query-dependent loss for web search, *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, ACM. pp. 141–150.

Buckley, C. (2009). Why current IR engines fail. *Information Retrieval* 12(6), 652–665.

Buckley, C., Dimmick, D., Soboroff, I., and Voorhees, E. (2006). Bias and the limits of pooling, *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Seattle, Washington, USA. pp. 619–620.

Buckley, C., and Voorhees, E.M. (2000). Evaluating evaluation measure stability, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Athens, Greece. pp. 33–40.

Callan, J. (2000). Distributed information retrieval, *Advances in Information Retrieval*. Springer US. volume 7 of *The Information Retrieval Series*, pp. 127–150.

Callan, J. (2012). The Lemur project and its ClueWeb12 dataset.

Callan, J., Hoy, M., Yoo, C., and Zhao, L. (2009). The ClueWeb09 dataset.

Carmel, D., and Yom-Tov, E. (2010). Estimating the query difficulty for information retrieval. Morgan & Claypool Publishers.

Carterette, B., Pavlu, V., Fang, H., and Kanoulas, E. (2009). Million Query Track 2009 Overview. Technical Report. National Institute of Standards and Technology.

Carterette, B., Pavlu, V., Kanoulas, E., Aslam, J.A., and Allan, J. (2008). Evaluation over thousands of queries, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Singapore, Singapore. pp. 651–658.

Chapelle, O., Metlzer, D., Zhang, Y., and Grinspan, P. (2009). Expected reciprocal rank for graded relevance, *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ACM, Hong Kong, China. pp. 621–630.

Clinchant, S., and Gaussier, E. (2009). Retrieval constraints and word frequency distributions: A log-logistic model for IR, *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ACM, New York, NY, USA. pp. 1975–1978.

Clinchant, S., and Gaussier, E. (2010). Information-based models for ad hoc IR, *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Geneva, Switzerland. pp. 234–241.

Clinchant, S., and Gaussier, E. (2011). Retrieval constraints and word frequency distributions a log-logistic model for IR. *Information Retrieval* 14(1), 5–25.

Collins-Thompson, K., Clarke, C.L., Bennet, P., Diaz, F., and Voorhees, E.M. (2014). TREC 2013 Web Track Overview. Technical Report. National Institute of Standards and Technology.

Collins-Thompson, K., Macdonald, C., Bennet, P., Diaz, F., and Voorhees, E.M. (2015). TREC 2014 Web Track Overview. Technical Report. National Institute of Standards and Technology.

Conover, W.J. (1999). Practical nonparametric statistics. 3rd ed., Wiley New York.

Cormack, G.V., Smucker, M.D., and Clarke, C.L.A. (2011). Efficient and effective spam filtering and re-ranking for large web datasets. *Information Retrieval* 14(5), 441–465.

Croft, B., Metzler, D., and Strohman, T. (2009). Search Engines: Information Retrieval in Practice. Pearson.

Cronen-Townsend, S., Zhou, Y., and Croft, W.B. (2002). Predicting query performance, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Tampere, Finland. pp. 299–306.

Dinçer, B.T., Macdonald, C., and Ounis, I. (2014). Hypothesis testing for the risk-sensitive evaluation of retrieval systems, *Proceedings of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Gold Coast, Queensland, Australia. pp. 23–32.

Dinçer, B.T., Macdonald, C., and Ounis, I. (2016). Risk-sensitive evaluation and learning to rank using multiple baselines, *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Pisa, Italy. pp. 483–492.

Fang, H., Tao, T., and Zhai, C. (2004). A formal study of information retrieval heuristics, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, United Kingdom. pp. 49–56.

Fang, H., Tao, T., and Zhai, C. (2011). Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.* 29(2), 7:1–7:42.

Ferro, N., Crestani, F., Moens, M.F., Leuven, K., Belgium, Silvestri, F., Kekäläinen, J., Rosso, P., Clough, P., Pasi, G., Lioma, C., Mizzaro, S., Maria, G., Nunzio, D., Hauff, C., Alonso, O., Yandex, P.S., Russia, and Silvello, G. (2016). Report on ECIR 2016: 38th european conference on information retrieval. *SIGIR Forum* 50(1), 12–27.

Geng, X., Liu, T.Y., Qin, T., Arnold, A., Li, H., and Shum, H.Y. (2008). Query dependent ranking using k-nearest neighbor, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Singapore, Singapore. pp. 115–122.

Gibbons, J.D., and Chakraborti, S. (2010). Nonparametric Statistical Inference. 5 ed., Chapman and Hall/CRC.

Guha, S., Rastogi, R., and Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. *SIGMOD Rec.* 27(2), 73–84.

Hanbury, A., Kazai, G., Rauber, A., and Fuhr, N. (2016). ECIR 2015: 37th european conference on information retrieval. *SIGIR Forum* 49(2), 36–46.

Harman, D., and Buckley, C. (2004). The NRRC reliable information access (RIA) workshop, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Sheffield, United Kingdom. pp. 528–529.

Harman, D., and Buckley, C. (2009). Overview of the reliable information access workshop. *Information Retrieval* 12(6), 615–641.

Hauff, C., Hiemstra, D., and de Jong, F. (2008). A survey of pre-retrieval query performance predictors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, ACM, Napa Valley, California, USA. pp. 1419–1420.

He, B., and Ounis, I. (2003). University of Glasgow at the Robust Track - A Query-based Model Selection Approach for the Poorly-performing Queries. Technical Report. National Institute of Standards and Technology.

He, B., and Ounis, I. (2004a). Inferring query performance using pre-retrieval predictors, *String Processing and Information Retrieval*. Springer Berlin Heidelberg. volume 3246 of *Lecture Notes in Computer Science*, pp. 43–54.

He, B., and Ounis, I. (2004b). A query-based pre-retrieval model selection approach to information retrieval, *Proceedings of the RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, Vaucluse, France. pp. 706–719.

Hiemstra, D. (2000). Using language models for information retrieval. Ph.D. thesis. University of Twente, Netherlands.

Hollander, M., and Wolfe, D.A. (1999). Nonparametric Statistical Methods. 2 ed., Wiley-Interscience.

Hummel, S., Shtok, A., Raiber, F., Kurland, O., and Carmel, D. (2012). Clarity re-visited, *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Portland, Oregon, USA. pp. 1039–1040.

Järvelin, K., and Kekäläinen, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.* 20(4), 422–446.

Jin, R., Falusos, C., and Hauptmann, A.G. (2001). Meta-scoring: Automatically evaluating term weighting schemes in IR without precision-recall, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New Orleans, Louisiana, USA. pp. 83–89.

Kang, I.H., and Kim, G. (2003). Query type classification for web document retrieval, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ACM, Toronto, Canada. pp. 64–71.

Kocabaş, I., Dinçer, B.T., and Karaoğlan, B. (2014). A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information Retrieval* 17(2), 153–176.

Kreyszig, E. (1970). Introductory Mathematical Statistics. John Wiley.

Krovetz, R. (1993). Viewing morphology as an inference process, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Pittsburgh, Pennsylvania, USA. pp. 191–202.

Kulkarni, A., and Callan, J. (2015). Selective search: Efficient and effective search of large textual collections. *ACM Trans. Inf. Syst.* 33(4), 17:1–17:33.

Lafferty, J., and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New Orleans, Louisiana, USA. pp. 111–119.

Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., Macdonald, C., and Vigna, S. (2016). Toward reproducible baselines: The open-source IR reproducibility challenge, *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*. Springer International Publishing, Cham, pp. 408–420.

Liu, T.Y. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval* 3(3), 225–331.

Lv, Y., and Zhai, C. (2012). A log-logistic model-based interpretation of TF normalization of BM25, *Proceedings of the 34th European Conference on Advances in Information Retrieval*, Springer-Verlag, Barcelona, Spain. pp. 244–255.

Macdonald, C., Santos, R.L., Ounis, I., and He, B. (2013a). About learning models with multiple query-dependent features. *ACM Trans. Inf. Syst.* 31(3), 11:1–11:39.

Macdonald, C., Santos, R.L.T., and Ounis, I. (2013b). The whens and hows of learning to rank for web search. *Information Retrieval* 16(5), 584–628.

Manmatha, R., Rath, T., and Feng, F. (2001). Modeling score distributions for combining the outputs of search engines, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, New Orleans, Louisiana, USA. pp. 267–275.

Manning, C.D., Raghavan, P., and Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

McDonell, R., Zobel, J., and Billerbeck, B. (2016). How informative is a term?: Dispersion as a measure of term specificity, *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Pisa, Italy. pp. 853–856.

Metzler, D., and Kurland, O. (2012). Experimental methods for information retrieval, *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Portland, Oregon, USA. pp. 1185–1186.

Peng, J. (2010). Learning to select for information retrieval. Ph.D. thesis. University of Glasgow.

Peng, J., He, B., and Ounis, I. (2009a). Predicting the usefulness of collection enrichment for enterprise search, *Proceedings of the 2nd International Conference on Theory of Information Retrieval: Advances in Information Retrieval Theory*, Springer-Verlag, Berlin, Heidelberg. pp. 366–370.

Peng, J., Macdonald, C., He, B., and Ounis, I. (2009b). A study of selective collection enrichment for enterprise search, *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ACM, Hong Kong, China. pp. 1999–2002.

Peng, J., Macdonald, C., and Ounis, I. (2010). Learning to select a ranking function, *Proceedings of the 32nd European Conference on Advances in Information Retrieval*, Springer-Verlag, Milton Keynes, UK. pp. 114–126.

Peng, J., and Ounis, I. (2009). Selective application of query-independent features in web information retrieval, *Advances in Information Retrieval*. Springer Berlin Heidelberg. volume 5478 of *Lecture Notes in Computer Science*, pp. 375–387.

Plachouras, V. (2006). Selective web information retrieval. Ph.D. thesis. University of Glasgow.

Plachouras, V., Cacheda, F., and Ounis, I. (2006). A decision mechanism for the selective combination of evidence in topic distillation. *Information Retrieval* 9(2), 139–163.

Plachouras, V., Ounis, I., and Cacheda, F. (2004). Selective combination of evidence for topic distillation using document and aggregate-level information, *Proceedings of the RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*, Vaucluse, France. pp. 610–622.

Ponte, J.M., and Croft, W.B. (1998). A language modeling approach to information retrieval, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Melbourne, Australia. pp. 275–281.

Porter, M.F. (1997). An algorithm for suffix stripping, *Readings in Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 313–316.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. (2007). Numerical Recipes 3rd Edition: The Art of Scientific Computing. 3 ed., Cambridge University Press, New York, NY, USA.

Ravana, S.D., and Moffat, A. (2008). Exploring evaluation metrics: GMAP versus MAP, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Singapore, Singapore. pp. 687–688.

Robertson, S. (2006). On GMAP: And other transformations, *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, ACM, Arlington, Virginia, USA. pp. 78–83.

Robertson, S., and Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3(4), 333–389.

Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple BM25 extension to multiple weighted fields, *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, ACM, Washington, D.C., USA. pp. 42–49.

Robertson, S.E. (1997). The probability ranking principle in IR. *Journal of Documentation* 33(4), 294–304.

Robertson, S.E., and Jones, K.S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science* 27(3), 129–146.

Robertson, S.E., and Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Springer-Verlag New York, Inc., New York, NY, USA. pp. 232–241.

Salton, G., and McGill, M.J. (1986). Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York, NY, USA.

Sanderson, M. (2010). Test collection based evaluation of information retrieval systems. *Foundations and Trends® in Information Retrieval* 4(4), 247–375.

Santos, R.L., Macdonald, C., and Ounis, I. (2010). Selectively diversifying web search results, *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ACM, Toronto, ON, Canada. pp. 1179–1188.

Shokouhi, M., and Si, L. (2011). Federated search. *Foundations and Trends® in Information Retrieval* 5(1), 1–102.

Shtok, A., Kurland, O., Carmel, D., Raiber, F., and Markovits, G. (2012). Predicting query performance by query-drift estimation. *ACM Trans. Inf. Syst.* 30(2), 11:1–11:35.

Si, L., and Callan, J. (2002). Using sampled data and regression to merge search engine results, *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Tampere, Finland. pp. 19–26.

Sullivan, D. (2015). Google still doing at least 1 trillion searches per year.

Tax, N., Bockting, S., and Hiemstra, D. (2015). A cross-benchmark comparison of 87 learning to rank methods. *Information Processing & Management* 51(6), 757 – 772.

Teevan, J., Dumais, S.T., and Liebling, D.J. (2008). To personalize or not to personalize: Modeling queries with variation in user intent, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Singapore, Singapore. pp. 163–170.

Tonellotto, N., Macdonald, C., and Ounis, I. (2013). Efficient and effective retrieval using selective pruning, *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, ACM, Rome, Italy. pp. 63–72.

Valizadegan, H., Jin, R., Zhang, R., and Mao, J. (2009). Learning to rank by optimizing NDCG measure, *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., pp. 1883–1891.

Voorhees, E.M. (2002). The philosophy of information retrieval evaluation, *Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems*, Springer-Verlag, London, UK. pp. 355–370.

Voorhees, E.M. (2004). Measuring ineffectiveness, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Sheffield, United Kingdom. pp. 562–563.

Voorhees, E.M. (2005). The TREC robust retrieval track. *SIGIR Forum* 39(1), 11–20.

Voorhees, E.M. (2007). TREC: Continuing information retrieval's tradition of experimentation. *Commun. ACM* 50(11), 51–54.

Voorhees, E.M., and Harman, D.K. (2005). TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing). The MIT Press.

Voorhees, E.M., Rajput, S., and Soboroff, I. (2016). Promoting repeatability through open runs, *Proceedings of the Seventh International Workshop on Evaluating Information Access*, Tokyo, Japan. pp. 17–20.

Wang, L., Bennett, P.N., and Collins-Thompson, K. (2012). Robust ranking models via risk-sensitive optimization, *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Portland, Oregon, USA. pp. 761–770.

White, R.W., Richardson, M., Bilenko, M., and Heath, A.P. (2008). Enhancing web search by promoting multiple search engine use, *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Singapore, Singapore. pp. 43–50.

Wurman, R.S. (2000). Information Anxiety 2. 2 ed., Que.

Yom-Tov, E., Fine, S., Carmel, D., and Darlow, A. (2005). Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval, *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Salvador, Brazil. pp. 512–519.

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

Zhao, Y., Scholer, F., and Tsegay, Y. (2008). Effective pre-retrieval query performance prediction using similarity and variability evidence, *Advances in Information Retrieval*. Springer Berlin Heidelberg. volume 4956 of *Lecture Notes in Computer Science*, pp. 52–64.

Zhou, Y., and Croft, W.B. (2007). Query performance prediction in web search environments, *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, Amsterdam, The Netherlands. pp. 543–550.

APPENDIX: ADDITIONAL TABLES

Table 1. Selective term-weighting result for ClueWeb{09A|12B} dataset (`NoAnchor`) over 281 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.21186	0	0.32909	0
ℵ†**SEL**	32.74	47.33	1	0.17363	1	0.29536	1
PL2 (c=8.0)	17.79	33.45	3	0.16737	2	0.28888	2
BM25 (k_1=1.6 b=0.2)	30.96	43.77	2	0.16511	3	0.28780	3
DFIC	12.81	27.76	5	0.16169	4	0.28406	4
RMLE	18.31	29.87	4	0.15637	5	0.27946	5
DPH	17.79	26.69	6	0.15477	6	0.27845	6
LGD (c=2.0)	7.12	19.93	8	0.14993	7	0.27292	7
RND	13.21	23.41	7	0.14905	8	0.27252	8
Dirichlet (μ=500)	8.54	18.51	9	0.14697	9	0.27006	9
DFRee	6.41	11.03	10	0.13311	10	0.25706	10
DLH13	5.69	7.47	11	0.11380	11	0.23755	11

Table 2. Selective term-weighting result for ClueWeb{09A|12B} dataset (`NoAnchor`) over 287 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.11572	0	0.24249	0
ℵ†PL2 (c=8.0)	19.86	37.98	3	0.09581	1	0.21886	1
ℵ†**SEL**	32.06	42.86	1	0.09381	2	0.21689	2
†DFIC	16.72	29.27	5	0.09003	3	0.21222	3
BM25 (k_1=1.2 b=0.3)	31.01	40.07	2	0.08901	4	0.21093	4
LGD (c=3.0)	6.97	17.07	8	0.08768	5	0.20908	5
RMLE	19.00	29.29	4	0.08727	6	0.20878	6
DPH	15.68	23.00	6	0.08593	7	0.20719	7
RND	13.35	21.89	7	0.08242	8	0.20276	8
Dirichlet (μ=500)	6.97	12.89	9	0.08090	9	0.20065	9
DFRee	5.57	9.76	10	0.07297	10	0.19051	10
DLH13	4.18	5.23	11	0.05678	11	0.16790	11

Table 3. Selective term-weighting result for Million Query 2009 dataset (`NoAnchor`) over 522 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.47048	0	0.49008	0
ℵ†**SEL**	27.39	41.76	1	0.40295	1	0.44966	1
DPH	20.88	33.33	3	0.39285	2	0.44235	2
PL2 (c=3.0)	8.62	21.65	8	0.38445	3	0.43790	3
BM25 (k_1=1.6 b=0.5)	26.25	37.74	2	0.38112	4	0.43629	4
LGD (c=2.0)	7.47	20.31	9	0.37789	5	0.43396	5
RMLE	17.20	27.95	4	0.37714	6	0.43379	6
RND	13.98	24.74	6	0.37518	7	0.43252	7
DFRee	12.84	22.22	7	0.37503	8	0.43199	8
DFIC	18.39	27.78	5	0.37127	10	0.43184	9
Dirichlet (μ=500)	5.75	17.62	10	0.37408	9	0.43155	10
DLH13	11.11	17.05	11	0.34443	11	0.41347	11

Table 4. Selective term-weighting result for Million Query 2009 dataset (`NoAnchor`) over 533 queries. Retrieval effectiveness is measured by statMAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	statMAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.34308	0	0.41788	0
ℵ†**SEL**	30.02	35.08	1	0.25077	1	0.35416	1
ℵ†DPH	17.07	24.39	3	0.24969	2	0.35241	2
ℵ†PL2 (c=8.0)	10.13	21.76	5	0.24520	3	0.34984	3
†DFRee	13.32	20.26	7	0.23861	4	0.34484	4
†LGD (c=5.0)	11.44	19.51	8	0.23813	5	0.34472	5
RMLE	16.68	23.06	4	0.23562	6	0.34292	6
BM25 (k_1=1.4 b=0.3)	28.33	32.83	2	0.23465	8	0.34278	7
RND	14.35	21.16	6	0.23510	7	0.34248	8
Dirichlet (μ=800)	10.51	15.76	11	0.23326	9	0.34102	9
DFIC	10.51	17.64	9	0.23258	10	0.34033	10
DLH13	13.51	17.26	10	0.20930	11	0.32346	11

Table 5. Selective term-weighting result for ClueWeb09A dataset (`NoAnchor`) over 188 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.24615	0	0.35650	0
ℵ†DFIC	23.94	40.43	2	0.19688	1	0.31498	1
ℵ†**SEL**	28.72	40.96	1	0.18836	2	0.30807	2
†DPH	18.62	30.32	4	0.18221	4	0.30214	3
†BM25 (k_1=1.0 b=0.4)	24.47	36.70	3	0.18291	3	0.30194	4
†PL2 (c=3.0)	7.45	25.53	6	0.18064	5	0.29909	5
RMLE	17.51	29.94	5	0.17897	6	0.29891	6
LGD (c=2.0)	4.26	20.21	9	0.17489	7	0.29451	7
RND	12.52	24.42	7	0.17228	9	0.29276	8
Dirichlet (μ=500)	6.38	21.28	8	0.17272	8	0.29254	9
DFRee	7.45	11.17	10	0.15380	10	0.27571	10
DLH13	7.45	9.57	11	0.13285	11	0.25628	11

Table 6. Selective term-weighting result for ClueWeb09A dataset (`NoAnchor`) over 194 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.15043	0	0.27715	0
ℵ†PL2 (c=8.0)	18.56	42.27	2	0.12483	1	0.24983	1
ℵ†**SEL**	32.99	43.81	1	0.12284	2	0.24800	2
†DFIC	19.07	33.51	4	0.11963	3	0.24527	3
BM25 (k_1=1.2 b=0.3)	28.35	37.63	3	0.11604	4	0.24088	4
RMLE	18.18	29.92	5	0.11279	5	0.23739	5
DPH	16.49	24.23	6	0.11110	6	0.23545	6
LGD (c=2.0)	5.67	13.40	9	0.10724	7	0.23093	7
RND	13.20	23.07	7	0.10624	8	0.23016	8
Dirichlet (μ=500)	5.15	13.92	8	0.10476	9	0.22809	9
DFRee	7.22	12.37	10	0.09358	10	0.21544	10
DLH13	5.15	6.70	11	0.07322	11	0.19045	11

Table 7. Selective term-weighting result for ClueWeb09B dataset (`NoAnchor`) over 190 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.27600	0	0.37522	0
ℵ†**SEL**	24.74	42.63	1	0.23193	2	0.34010	1
†LGD (c=18.0)	7.89	38.42	3	0.23197	1	0.33992	2
†PL2 (c=18.0)	11.58	38.42	4	0.23167	3	0.33938	3
BM25 (k_1=1.2 b=0.2)	24.21	38.95	2	0.22311	4	0.33288	4
†DFIC	11.05	26.84	9	0.22114	5	0.33116	5
Dirichlet (μ=2000)	8.95	31.05	7	0.21675	6	0.32803	6
RMLE	15.87	31.95	6	0.21011	7	0.32387	7
RND	13.19	30.22	8	0.20834	8	0.32241	8
DPH	22.11	32.63	5	0.20312	9	0.31987	9
DFRee	8.95	21.05	10	0.18058	10	0.30103	10
DLH13	10.00	14.21	11	0.15711	11	0.28109	11

Table 8. Selective term-weighting result for ClueWeb09B dataset (`NoAnchor`) over 192 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.15265	0	0.27825	0
ℵ†PL2 (c=12.0)	11.46	34.90	3	0.12975	1	0.25421	1
ℵ†LGD (c=8.0)	6.25	29.69	5	0.12642	2	0.25107	2
ℵ†**SEL**	30.21	41.15	1	0.12462	3	0.24960	3
†DFIC	14.06	26.56	7	0.12378	4	0.24837	4
†BM25 (k_1=2.0 b=0.2)	27.60	38.02	2	0.12174	5	0.24631	5
†Dirichlet (μ=1500)	9.38	26.04	9	0.11967	6	0.24410	6
RMLE	16.48	29.44	6	0.11645	7	0.24108	7
RND	12.89	26.44	8	0.11389	8	0.23845	8
DPH	18.23	31.77	4	0.11218	9	0.23703	9
DFRee	10.42	15.63	10	0.09889	10	0.22253	10
DLH13	6.25	9.38	11	0.07986	11	0.20004	11

Table 9. Selective term-weighting result for ClueWeb12-B13 dataset (`NoAnchor`) over 91 queries. Retrieval effectiveness is measured by NDCG100. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	NDCG100	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.12453	0	0.25053	0
ℵ†PL2 (c=10.0)	16.48	29.67	5	0.10483	1	0.22898	1
ℵ†**SEL**	34.07	43.96	1	0.10381	2	0.22823	2
ℵ†Dirichlet (μ=2000)	15.38	36.26	3	0.10367	4	0.22792	3
ℵ†LGD (c=5.0)	16.48	27.47	6	0.10377	3	0.22763	4
†RMLE	19.46	31.12	4	0.10024	5	0.22385	5
†BM25 (k_1=1.8 b=0.2)	29.67	42.86	2	0.09977	6	0.22338	6
ℵ†DPH	20.88	26.37	7	0.09928	7	0.22274	7
RND	13.80	23.74	8	0.09594	8	0.21886	8
DFRee	5.49	10.99	10	0.09108	9	0.21287	9
DFIC	4.40	13.19	9	0.09033	10	0.21239	10
DLH13	2.20	4.40	11	0.07474	11	0.19263	11

Table 10. Selective term-weighting result for ClueWeb12-B13 dataset (`NoAnchor`) over 93 queries. Retrieval effectiveness is measured by MAP. The models that are *not* statistically different ($p < 0.05$) from the selective approach (`SEL`) are marked with: † symbol according to the paired t-test and ℵ symbol according to the Wilcoxon signed-rank test.

	Accuracy %			Effectiveness		Robustness	
Model	$\sigma = 0$	$\sigma = 1$	Rank	MAP	Rank	GeoRisk	Rank
Oracle	100.00	100.00	0	0.04415	0	0.14909	0
ℵ†**SEL**	37.63	46.24	1	0.03551	1	0.13335	1
†PL2 (c=5.0)	13.98	26.88	6	0.03529	2	0.13277	2
ℵ†LGD (c=5.0)	19.35	33.33	4	0.03486	3	0.13205	3
ℵ†Dirichlet (μ=1500)	15.05	34.41	3	0.03432	4	0.13109	4
†BM25 (k_1=1.4 b=0.2)	38.71	44.09	2	0.03341	5	0.12930	5
†DPH	13.98	18.28	8	0.03341	6	0.12919	6
†RMLE	22.45	31.45	5	0.03333	7	0.12910	7
RND	14.22	22.29	7	0.03147	8	0.12537	8
DFRee	3.23	6.45	10	0.02998	9	0.12213	9
DFIC	8.60	12.90	9	0.02827	10	0.11884	10
DLH13	2.15	3.23	11	0.02249	11	0.10575	11

RÉSUMÉ

Name-Surname	: Ahmet Arslan
Foreign Language	: English
Birth Place and Year	: Muğla, Turkey / 1981
E-mail	: aarslan2@anadolu.edu.tr

Educational Status

- Anadolu University, Graduate School of Science, August 2016
  Doctor of Philosophy in Computer Engineering Department.
- Anadolu University, Graduate School of Science, July 2008
  Master of Science in Computer Engineering Department.
- Anadolu University, Faculty of Engineering, June 2004
  Bachelor of Science in Computer Engineering Department.

Career Experience

- Research Assistant, Anadolu University, Faculty of Engineering, 2005 - ∞
  Computer Engineering Department, Eskişehir, Turkey.

Administrative Tasks

- LLP/Erasmus Departmental Vice Coordinator, 2005 - 2010
  Anadolu University, Computer Engineering Department.

Articles Published in International Peer-review Periodicals

- Ahmet Arslan (2016), "DeASCIIfication approach to handle diacritics in Turkish information retrieval", Information Processing & Management, 52(2), pp. 326-339. doi: 10.1016/j.ipm.2015.08.004

Papers Submitted to International Meetings

- "Role of Apostrophes in Turkish Information Retrieval", International Conference on Software and Information Management, ICSIM 2012, Bangkok, Thailand, 24/11/2012
- "Frequent Pattern Mining Over Movie Plot Keywords", International Conference on Computer and Computational Intelligence, ICCCI 2011, Bangkok, Thailand, 03/12/2011
- "Automatic Grading System for Programming Homework", Computer Science Education: Innovation and Technology, CSEIT 2010, Phuket, Thailand, 06/12/2010
- "Quality Benchmarking Relational Databases and Lucene in the TREC4 Adhoc Task Environment", Computational Linguistics – Applications (CLA'10), Wisła, Poland, 20/10/2010 doi: 10.1109/IMCSIT.2010.5679643
- "An Approach to Prevent Stemming Side Effects in Information Retrieval", IADIS International Conference Applied Computing 2010, Timisoara, Romania, 16/10/2010
- "Turkish Text Retrieval Experiments Using Lemur Toolkit", IADIS International Conference Applied Computing 2009, Rome, Italy, 20/11/2009
- "Relational Databases versus Information Retrieval Systems: A Case Study", IADIS International Conference Applied Computing 2009, Rome, Italy, 20/11/2009
- "A Comparison of Relational Databases and Information Retrieval Libraries On Turkish Text Retrieval", International Conference on Natural Language Processing and Knowledge Engineering 2008 (NLP-KE'08), Beijing, China, 20/10/2008 doi: 10.1109/NLPKE.2008.4906748
- "TURKISH QUESTION ANSWERING: Question Answering for Distance Education Students", ICSOFT 2008, Porto, Portugal, 06/07/2008