Abstract of "Beyond keywords: finding information more accurately and easily using natural language" by Matthew Lease, Ph.D., Brown University, May 2010.

Information retrieval (IR) has become a ubiquitous technology for quickly and easily finding information on a given topic amidst the wealth of digital content available today. This dissertation addresses search for written and spoken natural language documents, including news articles, Web pages, and spoken interviews. Effective model estimation is identified as a key problem, and several novel estimation techniques are presented and shown to significantly enhance search accuracy.

While search is typically performed via a few carefully chosen keywords, formulating effective keyword queries is often unintuitive and iterative, particularly when seeking complex information. As an alternative to keyword search, this dissertation investigates search using "natural" queries, such as questions or sentences a person might naturally articulate in communicating their information need to another person. By moving toward supporting natural queries, the communication burden is shifted from user query formulation to system interpretation of natural language. The challenge in enacting such a shift is enabling automatic IR systems to more effectively cope with natural language. To this end, several new estimation techniques for modeling natural queries are described. In comparison to a maximum likelihood baseline, 15-20% relative improvement in mean-average precision (MAP) is demonstrated without use of query expansion.

When an IR system discovers or is provided one or more feedback documents exemplifying a user's information need, there is further opportunity to improve search accuracy by exploiting document contents for query expansion. However, since documents typically discuss multiple topics varying in importance and relevance to any information need, the system must again be able to effectively interpret verbose natural language. Consequently, an estimation method for leveraging such documents is presented and shown to yield state-of-the-art search accuracy. Depending on the base model employed, 15-85% relative MAP improvement is achieved.

When modeling higher-order lexical features or searching smaller document collections like cultural history archives, sparsity become particularly problematic for estimation. To cope with such sparsity, additional estimation methods are described which yield 5-20% relative improvement in MAP accuracy across varying conditions of query verbosity.

Beyond keywords: finding information more accurately and easily using natural language

by
Matthew Lease
B. S., University of Washington, 1999
Sc. M., Brown University, 2004

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2010

This dissertation by Matthew Lease is accepted in its present form by
the Department of Computer Science as satisfying the dissertation requirement
for the degree of Doctor of Philosophy.

Date _____ _____

Eugene Charniak, Director

Recommended to the Graduate Council

Date _____ _____

Mark Johnson, Reader

Date _____ _____

James Allan, Reader
University of Massachusetts Amherst

Approved by the Graduate Council

Date _____ _____

Sheila Bonde
Dean of the Graduate School

Acknowledgements

I am deeply grateful for the wonderful colleagues, friends and family that have been a constant source of inspiration, support, and kindness to me on the road toward my completing my doctoral studies. The path certainly had its ups and downs, bends and forks, and I was blessed to have had so many people pulling for me, providing assistance, and sharing good times with bad along the way. As I thank those who have helped or shared in the journey, I will certainly miss someone by accident, and I must beg forgiveness and understanding in this case.

From the Brown Laboratory for Linguistic Information Processing (BLLIP), I would like to first and foremost thank my advisor, Eugene Charniak. His encouragement, insights, patience, and sense of humor were endless as I discovered the field, explored topics, learned how to conduct research, and shared many laughs with him over the years. Unlike many of my graduate student peers, I never had to worry about whether or not I would be funded a given semester, even as my research interests shifted away from pure natural language processing (NLP) toward more applied information retrieval (IR). This allowed me to focus my time and energy on research instead of funding, which was a precious gift. In addition to Eugene, Mark Johnson was also there from the beginning with guidance, discussion, and good humor as a co-advisor. Our lab is fortunate to have not one, but two gifted and accomplished faculty members jointly overseeing all students' research in the lab, and Eugene and Mark have established an outstanding collaboration and lab environment, refining one another's ideas and shooting down any bad ones early. I would also like the many wonderful fellow students from the lab I have had the pleasure to get to know: Don Blaheta, Brock Pytlik, Heidi Fox, Yasemin Altun, Dmitriy Genzel, Ana Paula Simoes, Sharon Goldwater, Tori Sweetser, Tahir Butt, Alex Vasserman, Brendan Shean, Joseph Austerweil, Sarah Eisenstat, Lenora Huang, Stu Black, and Bevan Jones. I would like to especially thank David McClosky, Will Headden, and Jenine Turner; our semester working together in Prague was an amazing experience to have shared, and they have all been incredible supporters and friends. Special thanks also go to Keith Hall, Massimiliano Ciaramita, Hannah Rohde, David Ellis, Micha Elsner, and Engin Ural, all of whom have been terrific people to work and spend time with. Finally, I must thank BLLIP's one and only honorary member, Posh Spice, who inspired some of Eugene's best work (and laughs) over the years.

In addition to my great lab at Brown, I have the unusual pleasure of being able to thank a second set of remarkable colleagues and friends from another university, specifically those at the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts Amherst.

After stumbling upon an interesting IR paper that served as my primary introduction to the field (I must thanks its authors for providing an exciting paper for a non-IR practitioner) [Tao et al., 2006], two names I came across repeatedly in the important IR literature were James Allan and Bruce Croft at CIIR. Hence, when the time came to find an external committee member, contacting them was a natural choice, and James kindly agreed to serve in this capacity. In fact James went far above and beyond the typical responsibilities of a committee member.When I inquired about spending time regularly on-site at CIIR (thanks to some great encouragement from Steven Sloman), James immediately welcomed me into the lab, and his perspective, insights, and easy-going manner significantly contributed to my success. I also benefited greatly from discussions with Bruce Croft, who always had experience and wisdom to share. In addition to James and Bruce, the rest of the CIIR family welcomed me equally warmly. The students were outstanding: intelligent, motivated, passionate, and incredibly friendly. I take great pleasure in thanking Elif Aktolga, Niranjan Balasubramanian, Marc Cartright, Jeff Dalton, Van Dang, Henry Feild, Sam Huston, Jinyoung Kim, Jangwon Seo, Xiaobing Xue, and Xing Yi. Special thanks go to Michael Bendersky for his work helping inspire my own, his many valuable insights, and his endless good will. I would also like to thank Kate Moruzzi and Jean Joyce for their equally warm welcome and assistance, Andre Gauthier for technical assistance and fun chats, and David Smith for additional discussion and advising. I also had many great conversations with Manmatha. Finally, at conferences I had the further opportunity to get to know the extended CIIR family, and I cannot say enough good things about the many talented and kind people associated with CIIR, current and former members alike. Mark Smucker deserves special mention for his time and advice regarding my job search.

There are many wonderful people to thank in the wider Brown University community. Uğur Çetintemel has provided me with another great example of how to be an outstanding faculty member: teacher, advisor, friend, and generous benefactor (providing partial funding for me to attend my first conference while at Brown though the work was not affiliated with his own). Chad Jenkins, Shriram Krishnamurthi, and Meinolf Sellmann have also been friends and advisors over the years, and I had many interesting conversations with John Hughes. Technical staff did a fantastic job keeping the department up and running as well as being incredibly friendly and courteous, no many how many times user error was to blame: Jeff Coady, Mark Dietrich, John Bazik, Dorina Moulton, Phirum Peang, Max Silvas, and Kathy Kirman. Administrative staff were also incredible helpful and friendly: Lori Agresti, Katrina Avery, Lauren Clarke, Fran Palazzo, Suzi Howe, Jane McIlmail, and Dawn Reed. Special thanks go to Genie deGouveia, who always had a positive attitude and smile and laugh to share. I have known many great students in the Computer Science department who have positively impacted my time at Brown: Glencora Borradaile, Radu Jianu, Casey Marks, Victor Naroditskiy, Stefan Roth, and Frank Wood, to name a few. Extra thanks go to those truly dedicated friends who showed up on moving day: Casey, Engin, and Micha. Belinda and Claudia gave good advice and encouragement, and I was fortunate to have made some fantastic and dear friends who could really be counted on: Ines, Jean, Rita, Cecile, Zach, John, Elena, Marc-Andre, Robin, and Sabrina. Dan, Evan, and Will were fantastic flatmates and friends, and we had many memorable adventures with

the Plantations troop: Allison and Allison, Jen, Marian, Rebecca, and Shoshi. Thanks go as well to some fabulous visiting students who brought Europe to Brown: Jerome, Delphina, and Doreen. I also had an incredible time discovering the sport of recreational and competitive cycling thanks to Casey, Radu (and the nation of Romania), Jean, Giulia, Graeme, Kate, Graham, and others in the Brown Cycling Club and Refunds Now team. You each get ten Belgian points. Leslie, Mark, Jonathan, Allie, and Jose were great friends in the water, on the bike, and in the bagel shop. Lilly was a tremendous inspiration.

Doug Oard and Jimmy Lin were incredibly generous with their time and energy in offering feedback on my dissertation research, job search, and career going forward. Miles Efron also helped me with my job search, and Jian-Yun Nie has been very supportive and another source of inspiration. I would also like to thank faculty, staff, and students from four other universities where I made extended visits during my graduate studies. From the Signal, Speech and Language Interpretation (SSLI) Lab at the University of Washington, I would like to thank Mari Ostendorf for hosting me, as well as Jeremy G. Kahn and Dustin Hillard. Thanks to Mary Harper, I had the chance to participate in the summer workshop at Johns Hopkins University's Center for Language and Speech Processing (CLSP). Besides Mary, I benefited from working with, learning from, and/or getting to know Fred Jelinek, Sanjeev Khudanpur, Owen Rambow, Nizar Habash, Mona Diab, Roger Levy, Bonny Dorr, Yang Liu, Matt Snover, and Dan Melamed. At the Institute of Formal and Applied Linguistics (ÚFAL) at Charles University in Prague, the Czech Republic, I would like to thank Jan Hajič, Jaroslava Hlavácová, Kiril Ribarov, Silvie Cinková, Václav Novák, Pavel Pecina, Milan Fucík, Anna Kotešovcová, and Libuše Brdicková. Silvie has been an especially great supporter and source of inspiration. From the Spoken Language Systems (LSV) lab at Saarland University in Saarbrucken, Germany, I would like to thank Dietrich Klakow for being a wonderful and supportive host. Besides Dietrich, I would like to thank the many great students and staff at Saarland University who made me very welcome and provided many interesting discussions: Afra Alishahi, Andreas Bechorner, Mark Buckley, Grzegorz Chrupala, Bart Cramer, Rebecca Dridan, Friedrich Faubel, Antske Fokkens, Micha Jellinghaus, Judith Köhne, Dietmar Kuhn, John McDonough, Saeedeh Momtazi, Alexis Palmer, Barbara Rauch, Benjamin Roth, Diana Schreyer, Marianne Sottet, Michael Wiegand.

From Seattle, I would like to first thank Ira Kalet, who has been a tremendous mentor, friend, and supporter over the years, as well as inspiring me. Gaetano Borriello and Anthony LaMarca also served as great mentors who impressed me with their limitless energy, enthusiasm, and know-how. From the former LizardTech crew, I would like to thank John "Grizz" Deal, Vance Faber, Roland Sweet, Michael Gerlek, Jim White, and Peter Crook for giving me a great opportunity, experience, and exciting introduction to industrial research, as well as for their friendship. I would also like to thank Larry Ruzzo, Oren Etzioni, and Gaetano for providing outstanding undergraduate courses that also provided great introductions to research in their respective fields. Last but certainly not least, I would like to thank my dear friends from home who have been so supportive: Brice, Sonja, Katie, Shirley, Jake and Teresa, Konstantin and Kristi – you are the best!

Most of all, I would like to thank my family. My dear aunts Mary and Ellen have always been great

supporters from afar, and Shari, Chuck, Daniel Penn, and Jake, have provided great encouragement, advice, and welcome, as well as understanding my crazy schedule and brief visits. My grandparents, William and Betty, Lewis and Emily, provided a lifetime of love and encouragement continuing into my graduate studies. Kevin and Joelle have been there supporting me and cheering me on, and Camille and Zachary have been a tremendous joy to celebrate and help remind me of what matters most. Finally, no one has provided more support, encouragement, and love than my parents, Susan and Stephen. Words cannot say enough to thank them for all they have done and continue to do.

To my parents, Stephen and Susan

Contents

List of Tables

Chapter 1

Introduction

1.1 Motivation

Our information age has seen new, disruptive technological advances dramatically break down traditional barriers to communication, capture, and storage of information, with the effect that modern society has begun amassing vast stores of information at a tremendous rate in comparison to previous generations. As a natural consequence of such changes, computer and information sciences have emerged as key disciplines in organizing our vast and ever-growing collection of knowledge and enabling society's efficient operation and continuing progress.

Before us are a wide variety of challenges to and opportunities for improving support for information creation, storage, transmission, interpretation, access, use, enhancement, preservation, etc. This dissertation investigates one aspect of this broad picture: information retrieval (IR), i.e. enabling people to easily and accurately find and obtain existing information. Thanks to the Web revolution, the last decade has seen IR quickly enter into mainstream use as an essential and ubiquitous presence in our daily lives: we characterize the information to seek out in a repository such as the Web, and an automated system sifts through the repository to find the information that appears most relevant to our request. Given only the user's *query* and the *collection* of archived knowledge, retrieval accuracy ultimately depends on how well a system is able to interpret and match these two forms of information. When both consist entirely in natural (i.e. human) language content, retrieval accuracy becomes a question of system sophistication and accuracy in interpreting natural language.

This dissertation investigates estimation techniques for IR systems in order to improve their ability to interpret natural language and accurately retrieve desired information. In particular, we address search for natural language documents, such as news articles and Web pages, and present new estimation strategies that improve upon state-of-the-art document retrieval accuracy. A major contribution of this work is improved support for "natural" (a.k.a. verbose or long) queries in which the information being sought is described as if being explained to another person. While search using a few carefully selected keywords remains the dominant paradigm today, formulating

effective keyword queries is often unintuitive and iterative, particularly as information needs become more complex. In contrast, natural language provides the foundation of human communication and thereby supports easy, intuitive expression of arbitrarily complicated information needs. However, shifting the communication burden from user query formulation to system query interpretation is challenging in practice because current retrieval models perform poorly on verbose queries: a short keyword query typically yields greater retrieval accuracy despite being less informative. To overcome this deficiency, new estimation techniques are applied to specifically improve support for such queries.

We also investigate techniques for relevance and pseudo-relevance feedback (§2.5) in which the system is provided or finds (respectively) one or more documents exemplifying the information need. In this situation, there is further opportunity to improve retrieval accuracy by exploiting document contents. However, since documents typically discuss a variety of topics varying in importance and relevance to any particular information need, the system must again be able to effectively cope with verbose natural language evidence. Any uncertainty as to the relevance of example documents must also be managed. Consequently, accurate model estimation is once again paramount to achieving accurate retrieval. To meet this challenge, a new estimation strategy for leveraging such documents is presented and shown to perform as well or better than existing IR systems.

A final topic we investigate is spoken document search, specifically search for interviews from a cultural heritage archive which contain spontaneous speech. One of the challenges with this document collection is its small size, which leads to sparse statistics. This problem is exacerbated by modeling higher-order lexical features than simple unigram statistics. To address these challenges, we present a novel smoothing technique for bigram estimation and a method for leveraging external corpora for more robust estimation. Evaluation shows strong performance here as well.

1.2 Thesis Statement

This dissertation investigates the following thesis:

> Document retrieval accuracy can be significantly improved by better parameterizing existing retrieval models for "natural" queries and document feedback. Existing models can also be progressively augmented with additional features to enable incremental development and evaluation of richer query and document representations.

1.3 Contributions

Presented work supporting the thesis statement includes:

- Estimation methodology for natural queries (Ch. 3, Ch. 4, and Ch. 7)
- Estimation methodology for document feedback (Ch. 6)
- A learning framework for arbitrary feature-based parametric retrieval models (Ch. 3)

While the features developed and evaluated in Ch. 3 for query representation extend beyond simple word frequency statistics, the primary contribution made with regard to features stems from the learning framework's support for arbitrary features (in either the secondary feature set or the retrieval model itself) rather than for any specific features developed.

Contributions of the dissertation work are presented below under the categories of conceptual, methodological, and empirical. Contributions are first briefly listed with references to the corresponding section(s) describing the work. Following this, supporting details and background information are mentioned as the case merits.

1.3.1 Ideas

Conceptual contributions of the work include:

1. a feature-based interpretation of classic stochastic term-based search models
2. addressing estimation as a key challenge limiting search accuracy with natural queries

With regard to (1), explicitly distinguishing between the feature space employed and how feature weights are estimated enables us to separately examine and evaluate the contribution from each. How expressive is the feature space in its capacity to represent important differences between relevant and non-relevant documents? How challenging an estimation task does this feature space require, and how effective is the estimation technique employed? What new light do these questions shed on the relationship between different term-based search models and their particular strengths?

One fruit of adopting this perspective is recognizing that document-likelihood (§2.4.2) and query-likelihood (§2.4.3) are actually rank-equivalent models under equal parameterization (§4.4). This complements prior theoretical analysis which showed that while both approaches do model document relevance, query-likelihood may be more effective in practice due to addressing a simpler estimation problem (§2.4.3, [Lafferty and Zhai, 2003]).

Distinguishing features from estimation in term-based models also suggests the possibility of replacing one while preserving the other; we need not throw out the baby with the bath water. In particular, we might consider how the various estimation techniques being developed for supervised learning of feature-based *learning to rank* (LTR) models might be applied instead toward better estimating term-based models. Furthermore, it suggests an opportunity to empirically compare term-based and more general feature-based models on an equally-sound estimation footing to better understand the relative contributions being made by advances in feature development vs. estimation.

As for (2), the 2003 RIA Workshop (§2.7.2) presented a valuable taxonomy of errors over natural queries [Buckley and Harman, 2004]. Inspecting the distribution of errors across the taxonomy they developed suggested an even simpler bottom line than the taxonomy: failing to emphasize the "right" terms was the main problem across models, affecting approximately two-thirds of the queries they considered. In short, better term-weight estimation was needed. In a completely separate line of work, query-likelihood was theoretically shown to perform implicit maximum-likelihood (ML) estimation of term-weights in inferring the latent query unigram [Lafferty and Zhai, 2001]. The

connection between these lines of work lies in recognizing that ML estimates all query tokens as being equally important to the underlying information need, and while such an assumption is a reasonable approximation with keyword search, it is significantly at odds with the highly varying importance of terms in natural queries. It was therefore clear why standard query-likelihood tended to achieve less accurate search with natural queries relative to keyword queries: poor estimation. Moreover, theoretical [Zhai and Lafferty, 2004] and empirical [Fang et al., 2004] work demonstrating the close connection between query-likelihood and other term-based approaches [Buckley and Harman, 2004] suggested the problem was not limited to query-likelihood alone. While it has long been argued that assumptions like bag-of-words limit our ability to effectively model natural language, even the model of term interaction in Metzler and Croft's MRF approach (§2.6) embodied the same limiting ML assumption as the standard unigram by treating all observations as equally important (§4.2). Consequently, we saw an opportunity to improve search accuracy with these models by focusing on how to better estimate them.

1.3.2 Methods

Methodological contributions include:

1. a novel supervised learning framework for estimating any parametric retrieval model (Ch. 3)
2. supervised estimation of unigram query-likelihood search (Ch. 3)
3. supervised (Ch. 4) and feedback-based (Ch. 6) unigram estimation in the MRF model (§2.6)
4. blending explicit and pseudo-relevance feedback (PRF) (§2.5) with MRF modeling (Ch. 6)
5. Dirichlet-smoothed bigram modeling and "collection expansion" (Ch. 7)

Regarding (1), the learning framework exhibits several useful properties described in §3.4. Concrete application of these general ideas with regard to supervised unigram estimation is described for both query-likelihood in Ch. 3 (2) and the MRF model in Ch. 4 (3). We also show in Ch. 6 how MRF unigram estimation may be performed via feedback documents in the case of relevance feedback, as well as how such estimation can be further coupled with PRF estimation (4).

As for (5), a novel method for Dirichlet-smoothed bigram estimation and better estimating broad collection statistics is presented in Ch. 7. While this work is certainly not the first to suggest bigram modeling for IR (cf. [Song and Croft, 1999]), the formulation presented is the first we are aware of for combining bigram modeling with Dirichlet-smoothing. Similarly, while there has been previous work in expanding documents with similar ones found in external sources [Singhal and Pereira, 1999], there has been little work expanding collection-wide statistics via external corpora. One notable exception was work in topic detection and tracking, which leveraged external corpora to gain more robust statistics when only few documents had been seen [Allan et al., 1998].

1.3.3 Results

Empirical contributions include:

1. Improved search accuracy for news articles and Web pages using natural queries with unigram query-likelihood; oracle results show potential for greater accuracy (Ch. 3)
2. Improved search accuracy for news articles and Web pages using natural queries with MRF modeling; PRF yields further gains, and oracle results show additional potential (Ch. 4)
3. State-of-the-art search accuracy for Web pages given examples of known relevant pages by combining relevance and pseudo-relevance feedback with MRF modeling (Ch. 6)
4. Competitive search accuracy for spoken interviews using short, medium, and long queries with bigram query-likelihood (Ch. 7)

1.4 Information Retrieval Today

This section presents a brief and highly selective summary of IR today in order to establish current practice; more thorough introductions to IR are available elsewhere for the interested reader [Singhal, 2001, Zhai, 2007, Manning et al., 2008]. We begin by introducing the predominant approach to IR today, *bag-of-words* modeling, as well as reasons for its success. While surprisingly effective in practice, we illustrate the success of bag-of-words comes at the cost of model bias favoring succinct, keyword queries over more descriptive explanations of information needs. Next, we consider how the past decade's prevalence of end-user search engines built on this model may have significantly shaped how people have come to use and think about search today. In particular, we suggest widespread visibility of this particular search paradigm has led to an undesirable, perpetuating cycle between search engines and users: the latter write short keyword queries to fit the idiosyncratic behavior of search engines, and search engine behavior is optimized for such queries because that is what users tend to write, *ad infinitum*. Given this strong momentum in favor of keyword search, it is particularly telling to see that people nonetheless write descriptive, natural language queries when explaining information needs on Web Q&A sites like *Wondir* and *Yahoo Answers*. In fact, the growing traffic on these sites suggests that not only are people willing to write detailed queries in order to have their information needs satisfied, but also that Q&A sites have identified a legitimate market demand which existing search engines are failing to adequately address.

1.4.1 Bag-of-Words Modeling

Given that effectiveness of automated IR ultimately depends on depth of language understanding, it is remarkable how bag-of-words modeling has remained the predominant search paradigm into the present day. Be it vector similarity [Singhal et al., 1996], the probabilistic approach [Sparck Jones et al., 2000], or (typical) query-likelihood (§2.4.3), each adopts bags-of-words representation, employs

similar TF-IDF statistics [Zhai and Lafferty, 2004], and performs comparably in practice [Fang et al., 2004]. Under these models, archived documents (i.e. the granularity of information being sought) and user queries are represented by simple relative frequency statistics over the words; inter-word relationships are completely ignored. Despite its clearly limited capacity for modeling rich meaning, bag-of-words modeling has proven to be quite successful nonetheless. In order to understand this paradigm better, it is useful to examine its behavior and limitations. The most obvious and oft-cited criticism of bag-of-words is its complete disregard of modeling any form of interaction between terms: each word is considered in isolation when assessing the relevance of a given document to the query. Another simplifying modeling assumption is the standard models make no provision beyond relative frequency for inferring relative importance of query terms. The combined effect of these modeling assumptions has yielded IR systems biased toward succinct, keyword queries: additional terms introduced will tend to represent weaker correlations individually with the user's core information need. This means that despite being more informative to a human reader of the underlying information need, verbose queries tend to achieve lower search accuracy in practice than keyword queries [Zhai and Lafferty, 2002, Smucker and Allan, 2006, Kumaran and Allan, 2007, Bendersky and Croft, 2008]. It is important to recognize this system preference for succinct, keyword queries represents a significant departure from the form of language used by people to naturally communicate: whereas people intuitively employ more detailed descriptions to convey greater information, a user attempting to similarly provide a more informative request to his search engine today is likely to be rewarded by lower retrieval accuracy!

Fortunately, people are remarkably adept at recognizing the limitations of technology (if not the reasons for it) and adapting their behavior to accommodate it. In response to the system behavior described above, users have learned the importance of formulating their (potentially complex) information needs as keyword queries. While effective in the short term, this state of affairs is undesirable in several respects. First, keywords provide only limited capacity for expressiveness in comparison to unconstrained natural language, meaning it may not even be possible to express some information needs as effective keyword queries. Second, formulating an information need as an effective keyword query can be a challenging translation problem for users[1]. Third, keywords provide minimal context for query interpretation (i.e. inferring the information need underlying an observed query), reducing our potential to improve upon search engine accuracy both today and in the long-term. Finally, this dynamic between users and retrieval engines has produced an unfortunate perpetuating cycle: users write short keyword queries to fit this idiosyncratic system behavior, and IR research focuses on such queries because that is what users tend to write. Continuing this downward spiral, since succinct keyword queries provide minimal context for richer modeling, attempts at more sophisticated automatic language understanding rarely demonstrate benefit over bag-of-words. This, in turn, is often taken as further evidence that bag-of-words is a sufficiently accurate model of language use.

It is true that many attempts at more sophisticated modeling of keyword queries have failed

[1] While necessity and practice in recent years have honed users' skills in formulating keyword queries, this might not constitute the best example of laudable progress; it is a strange twist on life imitating art when people alter their natural use of language to suit a poor approximation of it, thereby "improving" model accuracy.

to demonstrate significantly improved retrieval accuracy relative to the additional computational burden they pose. In fact, despite the attention directed toward language understanding since the very beginnings of artificial intelligence (AI) as a field, deep understanding of language has remained rather elusive. When Karen Sparck Jones and several others independently wrote a collection of papers a decade ago reflecting on the observed and potential contribution of natural language processing (NLP) to IR [Hui, 1998, Lewis and Sparck Jones, 1996, Smeaton, 1999, Sparck Jones, 1997], their bleak observation might be best summed up in Smeaton's remark that "the impact of NLP on information retrieval tasks has largely been one of promise rather than substance" [Smeaton, 1999]. Nevertheless, it must be recognized that NLP has made significant strides, particularly over the past decade, in developing more effective and sophisticated models for other tasks, and the statistical revolution that swept through NLP over the past decade has yielded robust methodology for model estimation, inference, and evaluation. Moreover, by focusing IR efforts toward natural language queries rather than keywords, we remove the historical handicap on NLP and create an opportunity to exploit its potential more fully. Paraphrasing van Rijsbergen [Van Rijsbergen, 1979], the time appears ripe in light of these changes for another attempt at moving beyond bag-of-words toward richer language understanding for IR.

1.4.2 The Words out of the Bag: Q&A

While people generally think of search today in terms of keyword Web search, verbose Q&A search is also a significant and growing portion of today's search landscape. On Q&A sites, users post questions to be answered by other members of the user community (i.e. Q&A can be thought of as human-powered search). Assuming people would rather have their questions answered immediately then have to wait for others to respond, it is worth considering why people choose to use Q&A sites instead of more conventional automatic search. One possible explanation is that the information being sought simply does not exist on the web, in which case we might interpret these sites as providing access to an additional knowledge base. However, given the Web's existing vastness and enormous growth rate, this explanation seems less satisfying than it might have 10 years ago. Another argument against this explanation is that simple inspection shows some of these questions are in fact answerable via automatic search. For example, "Is [*sic*] there any other picture editing sites other than picnik and blingee???"[2]. Desire for social interaction might be another cause at work. But a more compelling explanation for use of Q&A over traditional search is difficulty with keyword query formulation: the user may have tried to use automatic search and failed to find an effective keyword query, or they may have not tried at all due to its expected difficulty or their preference for natural communication. Of course, as information needs become more complex, formulating queries in terms of keywords becomes increasingly difficult. For example, consider this brief excerpt from another user posting: "I'm pretty sure I am, or well WAS pregnant. [...] There are a lot od [*sic*] aspects to this question that I am going to lay out, [...]"[3]. In short, it appears Q&A sites address

[2]Posted on wondir.com, October 7, 2008

[3]Ibid.

a growing market need for better support of complex queries and ease of use. Furthermore, we see users willingly write detailed, informative queries when they believe such detail will enable them to obtain more satisfactory responses to their information needs.

It is certainly true that at one end of the spectrum, some information needs can be easily and naturally expressed in one or two keywords (e.g. a navigational query to locate a company's website), and query logs show (via minimal number of clicks and lack of query reformulation) that users leverage such keyword queries quite effectively in practice. We might assume keyword query formulation is not a heavy burden upon users in such cases of simple information needs. But it is also clear that there is another end of the spectrum, as we see with Q&A search. Recall earlier discussion regarding how users have adapted their use of search engines in response to their experience with what sort of queries yield successful searches (§1.4.1). This is characteristic of a more general language use phenomena in which speakers adapt their language for their intended recipient, are as verbose as needed to obtain the information being sought, and naturally express more complex information needs through more detailed, verbose descriptions. Generally speaking, describing more complex things requires more complex descriptions (e.g. think of information theory). Similarly, the extremes of navigational and Q&A search are not completely disconnected but represent two waypoints in a continuous spectrum of natural communication. People naturally expect to trade-off verbosity with effectiveness in conveying their intended meaning, and so it is problematic when search technologies operate according to a fundamentally different model of language use. It would be desirable if instead search engine interaction could become more consistent with human interaction to support more usable, intuitive formulation of queries.

Chapter 2

Background

2.1 Defining Relevance

The goal of Information Retrieval (IR) is to return information the user considers *relevant* to their request. Given the central role the notion of relevance plays in IR, it is worth saying something about what we mean by relevance. For the most part, we will follow the traditional practice of assuming relevance constitutes a binary measure over queries and documents: a document either contains some information relevant to a given query or it does not. While this simple distinction fails to model gradation in relatedness, it has nonetheless served for decades as a useful annotation standard supporting quantitative training and evaluation of retrieval systems, and it continues to provide a valuable foundation for developing new methodology.

2.2 Retrieval Scenarios

Ad hoc retrieval. In this task, the system is given a user "query" expressing an "information need" and a *collection* of documents in which to search for that information. We will assume with ad hoc retrieval that the system must infer the information need entirely on the basis of the query[1]. The output of search is a list of documents, ranked in order of (estimated) decreasing relevance, which the user may then peruse, use to refine his search, etc. Assuming the availability of a set of "canned" queries and corresponding human relevance assessments over the collection, the accuracy of a given system can be empirically evaluated and its strategies refined.

Relevance feedback. With relevance feedback (RF), the system is provided examples of relevant documents for a given query and must rank relevance of additional documents *on the same query*; for example, a user may indicate several documents relevant to his query in hopes of improving

[1] Today we are seeing increasing interest in user profiling via query logs and other means in order to better model a user's general and recent interests and thereby come by additional context for helping interpret and disambiguate new queries from the user. Similar prior context for query interpretation can be obtained via observing activities of others searchers to detect general trends, etc.

system accuracy in retrieving further documents. Our discussion of RF will assume a non-interactive setting in which example documents are provided up front along with the initial query[2]. RF is usually considered less typical than ad hoc retrieval since *explicit* feedback requires extra effort from the user in identifying relevant documents in addition to formulating a query. However, similar strategies like leveraging *implicit* relevance feedback garnered from query logs [Joachims, 2002] and pseudo-relevance feedback (PRF) [Lavrenko and Croft, 2001] are quite popular today with ad hoc retrieval. In §2.5, we further discuss the RF task and its methodology along with PRF.

Filtering. While we do not work on filtering in this dissertation, we mention it here for completeness and its close historical relation to ad hoc retrieval and RF. Filtering is an "online" classification task: documents are presented to the system in isolation to be classified as relevant or not to one or more standing queries (e.g. sorting incoming emails or news documents into different topical folders). Because documents are presented in isolation, the system must make an independent decision for each without considering it in the context of a collection other than those documents seen so far. As with RF, filtering has often been defined historically to assume the availability of example relevant documents alongside queries.

2.3 Controlled-vocabularies and stoplists

The question of whether to index all terms found in collection documents or only a subset has been a topic of interest for decades in the IR community. Reducing the size of the indexing vocabulary obviously has beneficial consequences for a system's storage requirements and its efficiency, and such reduction has been shown empirically often to have little cost or even benefit to overall retrieval accuracy across a variety of IR systems. Traditionally such vocabulary reduction is achieved by creating a simple, static list of terms to include (i.e. "a controlled-vocabulary") or exclude (i.e. a "stoplist" of "stopwords").

These two approaches can often be further distinguished via their treatment of open-class syntactic categories likes nouns and verbs: controlled-vocabularies devote significant attention to selection of open-class vocabulary while stoplists tend to be fairly conservative with regard to filtering out open-class terms. Partially this distinction is simply an inherent effect of selecting rather that excluding terms since open-classes are by definition far larger and harder to enumerate, as well as naturally changing across time, domains, and communities. The distinction between open-class and closed-class terms is of particular interest because open-class terms are generally viewed as more semantic or content-bearing and so more important for search. Of course closed-class terms also play an important role in conveying meaning. For example, the use of "very" in a hypothetical query "very hot days" could indicate user interest in more extreme or unusual weather than if the query were merely "hot days".

A limitation of employing a simple list of terms to include or exclude is that many terms are

[2] An alternative interactive setting could have a user incrementally indicate example documents in the course of inspecting returned results and iteratively refining his query, etc.

polysemous and have both open and closed-class senses. For example, "can" is both an auxiliary verb and a noun. There is also nothing preventing closed-class terms from being used as names (i.e. proper nouns), and the typical IR practice of case-folding to disregard differences in capitalization conflates these orthographically distinct cases in the indexing vocabulary. For example, "a" can be used a determiner or name of a vitamin (e.g. "Vitamin A"), "may" is both an auxiliary verb and name of a month or a person, etc. Note in a language like German where all nouns are capitalized, case-folding would impact cases like "can" as well.

An input query term is defined to be "out-of-vocabulary" (OOV) whenever either the term occurred in no document or the term did occur in some document but was excluded from indexing. Such OOV terms are typically simply ignored in query processing (as opposed to trying to map the OOV term to some indexed term via linguistic analysis, such as by orthographic or morphological similarity, or via an external resource beyond the index). Consequently, use of vocabulary reduction has the obvious drawback that some information will almost always be more easily or intuitively described using terms excluded from the index, making search for that information more difficult. If all query terms are OOV and therefore ignored, search must necessarily fail, and so more aggressive vocabulary reduction makes IR systems less robust, particularly when input queries are short.

Historically, use of controlled vocabularies was motivated both by storage and efficiency limitations of early IR systems as well as reflecting an influence from library science in which manual indexing had played a crucial role in taxonomically organizing knowledge sources to support structured and efficient access. However, as IR systems matured in ranking sophistication and indexing all terms became practical even for large text sources, empirical studies began to show that systems indexing all open-class terms were generally just as effective as those employing controlled vocabularies as well as enabling more flexible access. Consequently, use of controlled vocabularies has largely receded from common use in today's IR systems.

However, use of stoplists has remained quite popular into the present day, at least in academic research, though of course exceptions exist (cf. [Fang et al., 2004, Mei et al., 2007, Zhai and Lafferty, 2002]). As an illustrative example of stopping, consider the 418 term stoplist employed by the INQUERY system [Allan et al., 2000] that was carefully developed over the course of participating in multiple TREC evaluations. In the following query (TREC topic 705's `description`), words appearing in INQUERY's stoplist are marked by underlining: *Identify any efforts, proposed or undertaken, by world governments to seek reduction of Iraq's foreign debt.* While terms stopped in this example are all from closed-class categories, the INQUERY stoplist also includes a variety open-class terms such as *nowadays, seeing, slept, smoke, spat, ...* As mentioned earlier, use of stopping can be quite detrimental in some cases. For example, in the query "smoke signals", "smoke" plays a critical role in conveying the query's meaning, yet it is removed by INQUERY's stop list. Retrieval fails entirely for TREC keyword queries "who and whom" (topic 531) and "May Day" (topic 803) in which all query terms appear in the stop list.

So while use of stoplists offers storage and efficiency savings, there is no free lunch. For every word in natural language (that could potentially be stopped), one can imagine a query for which

that word would play an important role in conveying the query's intended information need. In other words, use of stopping inherently reduces system robustness, and so stoplist use represents an implicit tradeoff between robustness and other aspects of system behavior being optimized. As such, it is important to recognize cases of this tradeoff at work whenever stopping is employed.

2.4 Modeling Paradigms

2.4.1 Vector Similarity

A classic and still competitive bag-of-words approach to IR is the vector space model. In this approach, the query and documents are represented as vectors over the collection vocabulary and documents are ranked on the basis of vector similarity [Singhal et al., 1996, Singhal, 2001]. Several key statistics are utilized in this model: term frequency (TF), inverse document frequency (IDF), and length normalization. TF is a measure of term salience: the more often a query term occurs in a document, the more information the document is assumed to contain related to that term. Since terms will tend to occur more frequently in longer documents regardless of topic, document length normalization is usually applied to remove this bias: relative term frequencies are used in place of absolute counts. IDF measures term importance: a query term occurring rarely in the collection is assumed to be more useful in discriminating between documents than a determiner like "the" which likely occurs in every document. In addition to TF and IDF statistics, document length has also been heavily exploited to improve retrieval accuracy. For example, pivoted document length normalization applies variable (non-Euclidean) normalization to correct for observed error between estimated relevance under standard normalization and actual relevance values observed on development data [Singhal et al., 1996].

2.4.2 Document-Likelihood

Like the vector-similarity approach, document-likelihood (also known as Okapi or "the probabilistic approach")[3] represents another bag-of-words approach with a long and influential history of strong empirical performance. Much of the derivation presented below follows an earlier presentation [Lafferty and Zhai, 2003]. Document-likelihood ranking is based on Robertson's famous probability ranking principle (PRP), which showed that optimal system behavior under several evaluation metrics such as expected average precision could be achieved by ranking documents according to the probability of their belonging to the relevant class [Robertson, 1977]. Assuming queries and documents are represented by random variables Q and D respectively, with a binary random variable R indicating relevance r or non-relevance $\bar{r}$, Robertson argued for ranking documents by their posterior probability of relevance $P(R = r|Q, D)$. Note that if our goal were to classify documents as relevant

[3] While not the earliest probabilistic model for IR [Maron and Kuhns, 1960], Okapi came to be known as the probabilistic approach due to its influential impact and to distinguish it from (non-probabilistic) vector similarity. We adopt "document-likelihood" from [Lafferty and Zhai, 2003] to emphasize the close relationship between this approach and query-likelihood (§2.4.3).

or non-relevant, as done in filtering (§2.2), rather than rank them, this posterior would also define the Bayes optimal decision criterion (i.e. would choose class assignments to minimize the probability of error).

Next, it can be seen that ranking documents by $P(R = r|Q, D)$ is equivalent to ranking them by the likelihood ratio between the competing hypotheses of relevance and non-relevance:

$$p(r|Q, D) \stackrel{rank}{=} \frac{p(r|Q, D)}{p(\bar{r}|Q, D)} \tag{2.4.1}$$

where $\stackrel{rank}{=}$ denotes rank-equivalence. To see this, consider generic probabilities p and $1 - p$ and consider the interval $[0, 1]$ over which they are defined. The two functions are strictly increasing and decreasing, respectively, and therefore their ratio $\frac{p}{1-p}$ will also be strictly increasing. Since p and $\frac{p}{1-p}$ are both strictly increasing over the same interval, they are therefore rank-equivalent.

Rather than estimate relevance directly, as done with *learning to rank* [Joachims et al., 2007], the probabilistic approach instead adopts a generative approach via application of Bayes' Rule:

$$\frac{p(r|Q, D)}{p(\bar{r}|Q, D)} = \frac{p(Q, D|r)p(r)}{p(Q, D|\bar{r})p(\bar{r})} \tag{2.4.2}$$

Note that ranking by the ratio conveniently avoids computation of the marginal $P(Q, D)$.

The key step in the next portion of the derivation is that the posterior joint probability $P(Q, D|r)$ is factored by generating first the query Q and then the document D conditioned on the query. The last step of proportionality is justified by $p(R, Q)$ being constant for all documents with regard to the same query.

$$\begin{aligned} \frac{p(r|Q, D)}{p(\bar{r}|Q, D)} &= \frac{p(D|Q, r)p(Q|r)p(r)}{p(D|Q, \bar{r})p(Q|\bar{r})p(\bar{r})} \\ &= \frac{p(D|Q, r)p(r, Q)}{p(D|Q, \bar{r})p(\bar{r}, Q)} \\ &\propto \frac{p(D|Q, r)}{p(D|Q, \bar{r})} \end{aligned} \tag{2.4.3}$$

Equation (2.4.3) shows how the document-likelihood model gets its name: given a query, the distributions $p(D|Q, r)$ and $p(D|Q, \bar{r})$ characterize the space of documents likely to be relevant and non-relevant.

Next, the bag-of-words assumption is adopted to actually generate the documents:

$$p(D|Q, R) = \prod_{w \in D} p(w|Q, R)$$

for both $R = r$ and $R = \bar{r}$.

Finally, note the above model requires us to know whether or not D is relevant to Q in order to estimate the distributions $p(D|Q, R)$. Given examples of documents relevant and not-relevant to Q, the corresponding unigram distributions $p(w|Q, r)$ and $p(w|Q, \bar{r})$ above can be estimated and used to compute a document's likelihood under each of the competing relevance hypotheses, r and $\bar{r}$, in order to rank documents. However, while the model appears well-suited to the task of relevance

feedback, it is unclear how to proceed with estimation in the case of ad hoc retrieval (§2.2). While we could estimate $P(w|Q,R)$ from the query by maximum likelihood (ML), this would be error-prone since a typical succinct query provides little context for estimating unigram parameters over the entire vocabulary. Instead, $p(w|Q,\bar{r})$ is often estimated by assuming all documents are non-relevant [Lafferty and Zhai, 2003] and $p(w|Q,r)$ is usually assumed to be uniform (i.e. parameters are constant). Letting k denote this constant and C denote the collection of documents, these assumptions yield:

$$\begin{aligned} \frac{p(r|Q,D)}{p(\bar{r}|Q,D)} &\propto \frac{p(D|Q,r)}{p(D|Q,\bar{r})} \\ &\equiv \frac{k}{p(D|Q,C)} \\ &\overset{rank}{=} -p(D|Q,C) \end{aligned} \tag{2.4.4}$$

From which documents can then be ranked for ad hoc retrieval.

The well-established probabilistic model Okapi BM25 operates within the framework laid out above and has been carefully refined over many years of participation in TREC evaluations [Sparck Jones et al., 2000, Fang et al., 2004]. In addition to leveraging basic TF-IDF statistics in estimating the above probability distributions, BM25's probability model also incorporates average document length, provides several free parameters for tuning on development data, and facilitates query term weighting, which has been shown to be useful with longer queries.

2.4.3 Query-Likelihood

A decade ago, Ponte and Croft proposed a new paradigm for IR based on language modeling [Ponte and Croft, 1998]. In this paradigm, one assumes a latent language model (LM) underlies each observed document and infers the relevance of each document by the posterior probability of observing the query as a random sample generated by each document's underlying LM. As this description suggests, the original derivation of the language modeling approach forwent the explicit notion of relevance on which the document-likelihood approach (§2.4.2) was derived , instead modeling a connection between observed queries and latent document models. We will refer to this approach as "query-likelihood" rather than "language modeling" to make explicit that the approach is query-generative and to show its close relationship with document-likelihood. The key challenges in this approach are hypothesizing the form of the underlying source models and finding an effective estimation procedure given the brevity of observed evidence. A strength of the approach lies in the pre-existing theoretical foundation for general language modeling and set of proven estimation techniques developed by earlier work in speech recognition (and more recently, machine translation). Query-likelihood has been shown to have a strong theoretical connection to classic TF-IDF statistics [Zhai and Lafferty, 2004] and perform comparably to both vector space (§2.4.1) and document-likelihood §2.4.2 approaches in practice [Fang et al., 2004].

Relevance-agnostic Derivation

The LM approach defines a query-generating process; since queries are traditionally rather brief in comparison to documents, generating queries rather than documents provides a firmer foothold for statistical estimation. Since the goal is to rank documents for a given query, Bayes' rule is invoked to indirectly model document likelihood via a direct model of query likelihood. If we (1) assume all documents are equally likely to be relevant *a priori* and (2) ignore the prior over queries $P(Q)$ which is constant when ranking document relevance to Q, the LM approach can be expressed succinctly as:

$$p(D|Q) = \frac{p(Q|D)p(D)}{p(Q)} = \frac{p(Q|D)}{p(Q)} \stackrel{rank}{=} p(Q|D) \tag{2.4.5}$$

As in the case of the probabilistic method above, the bag-of-words assumption is usually adopted to generate the query from a unigram model:

$$p(Q|D) = \prod_{w \in Q} p(w|D) \tag{2.4.6}$$

In other words, we compute query likelihood by the product of individual term probabilities under the document LM $P(\cdot|D)$.

Relevance-based Derivation

It was subsequently shown that the language modeling approach could be derived from the same explicit notion of relevance on which the probabilistic approach was based [Lafferty and Zhai, 2003], establishing an important connection by showing both approaches can be interpreted within the same probabilistic framework and merely represent differences between their independence assumptions and estimation procedures. The connection also has useful implications for relevance modeling under the language modeling paradigm. Our presentation below follows one given earlier [Lafferty and Zhai, 2003].

Beginning with Equation (2.4.2) in log form, we follow the same series of steps as used in deriving Equation (2.4.3) except the posterior joint probability $P(Q, D|r)$ is factored in the opposite order by generating first the document D and then the query Q:

$$\begin{aligned} log\ \frac{p(r|Q,D)}{p(\bar{r}|Q,D)} &= log\frac{p(Q,D|r)p(r)}{p(Q,D|\bar{r})p(\bar{r})} \\ &= log\frac{p(Q|D,r)p(D|r)p(r)}{p(Q|D,\bar{r})p(D|\bar{r})p(\bar{r})} \\ &= log\frac{p(Q|D,r)p(r|D)}{p(Q|D,\bar{r})p(\bar{r}|D)} \\ &= log\frac{p(Q|D,r)}{p(Q|D,\bar{r})} + log\frac{p(r|D)}{p(\bar{r}|D)} \end{aligned}$$

Where the final term indicates a query-independent document prior of relevance. Next, recall the semantics of non-relevance: $R = \bar{r}$ indicates D is not related to Q with respect to the latent information need underlying Q. Given this, let us make an assumption that Q and D are completely

independent when $R = \bar{r}$. The idea here is that since we know D and Q are at least unrelated with respect to this information need, we take a leap of faith that they are sufficiently unrelated in general that modeling them as being completely independent will be a reasonable approximation. This assumption reduces $p(Q|D, \bar{r})$ to $p(Q|\bar{r})$, which for a given fixed Q is constant across documents being ranked and so can be ignored.

$$\begin{aligned} log\ \frac{p(r|Q,D)}{p(\bar{r}|Q,D)} &= log\ \frac{p(Q|D,r)}{p(Q|\bar{r})} + log\frac{p(r|D)}{p(\bar{r}|D)} \\ &\stackrel{rank}{=} log\, p(Q|D,r) + log\frac{p(r|D)}{p(\bar{r}|D)} \end{aligned} \tag{2.4.7}$$

Next, consider an additional assumption that D and R are also independent; this assumption embodies the idea that documents and users' information needs arise independently of one another. Ignoring a notion of shared latent topics underlying documents and queries, this assumption models a generative process in which documents are written without foresight of future information needs and users' information needs arise due to external factors rather than based on the set of available documents. Adopting this assumption, the document prior above no longer depends on D and so becomes a constant factor with regard to ranking and can be therefore ignored:

$$\begin{aligned} log\ \frac{p(r|Q,D)}{p(\bar{r}|Q,D)} &\stackrel{rank}{=} log\, p(Q|D,r) + log\frac{p(r|D)}{p(\bar{r}|D)} \\ &= log\, p(Q|D,r) + log\frac{p(r)}{p(\bar{r})} \\ &\propto log\, p(Q|D,r) \end{aligned} \tag{2.4.8}$$

Whereas we saw earlier that the probabilistic approach can be understood as a *document-generating* or *document-likelihood* model; Equation (2.4.8) shows the language modeling approach instead generates queries, defining a *query-likelihood* model. Said another way, the language modeling approach characterizes the space of possible queries for which a given document is likely to be relevant. As in Equation (2.4.6), we can once again assume the queries generated from a bag-of-words unigram model:

$$p(Q|D,r) = \prod_{w \in Q} p(w|D,r) \tag{2.4.9}$$

Finally, note this relevance-based derivation of the language modeling approach leaves it with the same dilemma faced by the probabilistic approach: we must know whether or not D is relevant to Q in order to estimate the unigram model underlying Q. As with the probabilistic approach, and additional simplifying assumption must be made: we assume that queries do not arise from user information needs but simply as random samples from documents, i.e. that Q depends only on D and not R. As a consequence, $p(Q|D, r)$ is further reduced to $p(Q|D)$, connecting the relevance-based derivation to the original language model formulation presented in Equation (2.4.5). However, there is an important distinction to note here in comparing the probabilistic and language modeling approaches. In the case of the former, examples of known relevant documents for Q enable us to better estimate the likelihood of observing a given document under the competing hypotheses of

relevance and non-relevance and so better rank documents for Q. Here, however, this is not the case: as a query-generating model, knowing D is relevant to Q enables us to better model queries to which D would be relevant but does not improve our ability (at least directly) to rank other documents for Q. Another way to see this is that since the language modeling approach involves estimating a different query-likelihood model conditioned on each document, any relevance information provided regarding a particular document only improves our ability to better estimate the document-specific model to which the relevance example directly pertains.

Comparison to Document Likelihood

The last point above highlights that important differences exist between document-likelihood and query-likelihood approaches that bear consideration in comparing the merits of each:

- in the absence of relevance feedback, query-likelihood provides a better statistical foothold for estimation since documents tend to be significantly longer than queries
- given relevance feedback for a query Q, document-likelihood provides a direct means for improved model estimation and thereby better ranking document relevance to Q
- whereas query-generation requires comparing the likelihood of the same, fixed-length query under different document models, document-generation requires length-normalization since longer documents are necessarily less probable
- query-generation is less sensitive to error introduced by strong independence assumptions like bag-of-words since queries are shorter than documents
- the document prior in query-generation (Equation 2.4.7) provides an opportunity to model document aspects such as length and hyperlink structure indicative of a document's prior probability of relevance across queries [Richardson et al., 2006]

Smoothed Document Unigram Estimation

In query-likelihood, how do we estimate the latent document unigram $P(\cdot|D)$ we postulate as underlying each observed document in the collection? One option is maximum-likelihood (ML). Assuming vocabulary size V, word w_i occurring in D with frequency f_{w_i}, and $P(\cdot|D)$ being parameterized by Θ, we could seek the particular $\widehat{\Theta}$ maximizing D's likelihood

$$P(D|\Theta) = \prod_{i=1}^{V} \theta_i^{f_{w_i}} \tag{2.4.10}$$

which would be the assignment to Θ respecting the empirical frequencies f. However, such use of ML is problematic in that a single unobserved query term would completely nullify query likelihood, making the entire framework exceedingly fragile. The problem here is that in observing only a small sample (i.e. a brief document) from an underlying distribution, effects of chance variation will be prominent and distort sample statistics away from those governing the generating distribution.

Instead one commonly employs smoothing to discount the probability mass assigned to observed terms and reserve some probability mass for all unseen terms. The most common practice is to estimate the document model as a mixture between ML estimates from the observed document and the collection of all documents:

$$p(Q|D,C) = \prod_{w \in Q} \lambda\, p(w|D) + (1-\lambda) p(w|C) \tag{2.4.11}$$

A principled way to accomplish this is via *maximum a posteriori*[4] estimation in the Bayesian framework by treating collection statistics as a prior over the document unigrams. *A priori*, we might reasonably assume $P(\cdot|D)$ should resemble the collection's *average* document model $P(\cdot|C)$. This, in turn, could be estimated via ML by summing statistics across all documents, which would provide sufficient evidence for a much more robust estimate.

Such prior knowledge can be elegantly incorporated into a language model via the Dirichlet distribution, specified by hyper-parameters $\alpha > 0$ and defining a distribution over multinomial parameterizations $P(\Theta;\alpha)$ [MacKay and Peto, 1995]. For the unigram model defined above, the corresponding Dirichlet prior would be defined as

$$P(\Theta;\alpha) \doteq Dir(\alpha) = \frac{1}{Z(\alpha)} \prod_{i=1}^{V} {\theta_i}^{\alpha_i - 1} \tag{2.4.12}$$

where $Z(\alpha)$ denotes normalization. This prior is particularly convenient for *maximum a posteriori* estimation because its distribution is conjugate to the multinomial, meaning the posterior will also be Dirichlet. Hence, combining likelihood (2.4.10) and prior (2.4.12):

$$P(\Theta|D;\alpha) \propto P(\Theta;\alpha) P(D|\Theta) \propto \prod_{i=1}^{V} {\theta_i}^{\alpha_i - 1} \prod_{i=1}^{V} \theta_i^{f_{w_i}} = \prod_{i=1}^{V} \theta_i^{f_{w_i} + \alpha_i - 1} \tag{2.4.13}$$

A true Bayesian would next compute the predictive distribution over Θ, but we will instead assume a peaked posterior and find the single most-likely $\widehat{\Theta}$ to explain our data via the maximum approximation. Comparing our likelihood and posterior equations (2.4.10) and (2.4.13), we can see that maximizing the posterior is quite similar to maximizing the likelihood, only the data now consists of both the empirical evidence and "pseudo" α observations. In other words, the posterior maximum is simply the combined relative frequency of the observed and pseudo data. Finally, letting $\alpha - 1 = \mu P(\cdot|C)$ for $\mu >= 0$, we see our empirical document statistics are smoothed with μ pseudo-counts drawn from our average document model $P(\cdot|C)$ to yield IR's canonical Dirichlet-smoothed unigram model [Zhai and Lafferty, 2004]:

$$P(w|D,C) = \frac{f_w + \mu P(w|C)}{|D| + \mu} \tag{2.4.14}$$

where $|D|$ indicates D's length and the μ hyper-parameter expresses strength of the prior in smoothing. Correspondence with Equation 2.4.11 is shown by expressing λ as a function of μ and $|D|$.

$$\lambda = \frac{|D|}{|D| + \mu} \tag{2.4.15}$$

[4] While the acronym MAP is often used in this context with the statistical Bayesian literature, we use MAP exclusively in referring to the "mean-average precision" metric.

The intuitive attractiveness of this smoothing strategy lies in the fact that as document length increases, providing more evidence for the ML estimate, the impact of the prior model will correspondingly diminish. Despite the elegance of this Bayesian intuition, however, subsequent work has indicated the real practical benefit of Dirichlet smoothing arises not from better estimation, but rather from Dirichlet smoothing's implicit incorporation of length-normalization to bias retrieval in favor of longer documents [Smucker and Allan, 2005]. It may also suggest why more rigorous Bayesian estimation of the predictive distribution failed to improve retrieval accuracy vs. use of the simpler point estimation [Zaragoza et al., 2003]. Consequently, the continuing popularity of Dirichlet smoothing might be interpreted as a reflection of its simplicity in incorporating length-normalization into smoothing as well as its consistently strong empirical performance.

Equivalence with KL-divergence Ranking

Given an input query $Q = q_1 \dots q_m$, query-likelihood infers D's relevance to Q as the probability of observing Q as a random sample drawn from Θ^D. If we assume bag-of-words modeling, Θ^D specifies a unigram distribution $\{\theta^D_{w_1} \dots \theta^D_{w_N}\}$ over the document collection vocabulary $V = \{w_1 \dots w_N\}$. Letting f^Q_w denote the frequency of word w in Q, query likelihood can be expressed in *log* form as:

$$log\, p(Q|\Theta^D) = \sum_{i=1}^{m} log\, \theta^D_{q_i} = \sum_{w \in V} f^Q_w \; log\, \theta^D_w = f^Q \cdot log\, \theta^D \tag{2.4.16}$$

While this formulation is completely valid, it is somewhat cumbersome to work with in that the relative importance of query terms can only be expressed via their relative frequency in the query string, meaning long, complex query strings would be required to express any fine-grained distinctions in term importance. Fortunately, we may arrive at an equivalent, more expressive generalization by revising our formulation to explicitly model the user's information need [Lafferty and Zhai, 2001]. Specifically, we assume the observed Q is merely representative of a latent query model parameterized by $\Theta^Q = \{\theta^Q_{w_1} \dots \theta^Q_{w_V}\}$, consistent with the intuitive notion that the underlying information need might be verbalized in other ways besides Q. We can re-express query likelihood in terms of Θ^Q's maximum-likelihood (ML) estimate $\widehat{\Theta^Q} = \frac{1}{m} f^Q$ as:

$$log\, p(Q|\Theta^D) = f^Q \cdot log\, \theta^D = m\, \widehat{\Theta^Q} \cdot log\, \theta^D \stackrel{rank}{=} -D(\widehat{\Theta^Q}||\Theta^D) \tag{2.4.17}$$

This derivation shows that inferring document relevance on the basis of Q's likelihood given Θ^D has an alternative explanation of ranking on the basis of minimal KL-divergence between Θ^Q and Θ^D assuming Θ^Q is estimated by ML. This insight is useful because it transforms the task of optimal query formulation into one of optimal query model estimation, suggesting how search accuracy could be improved via more effective estimation of Θ^Q.

2.5 Relevance & Pseudo-relevance Feedback

Input queries are non-optimal; information is often lost as a user formulates his information need into a concrete query for input to the system (e.g. due to brevity, ambiguity, miscommunication,

...). Consequently, there is often a paraphrase mismatch in how a query and its relevant documents refer to the same information. However, if we had some additional source of knowledge regarding the information need in addition to the query, we could exploit that as well to better infer what information the user desires.

Relevance feedback (RF) refers to a retrieval scenario in which the system is provided not only with the query, but also with examples of relevant (and possibly non-relevant) documents (§2.2). This allows document contents to be leveraged alongside the query in inferring the user's information need. For example, a user may explicitly indicate several documents relevant to his query in hopes of raising the system's search accuracy in finding additional relevant documents. The system, in turn, might harvest terms from those documents and use them to augment the input query. In general, our use of "relevance feedback" will refer to both this task and the methodology for using feedback documents in conjunction with the query.

Pseudo-relevance feedback (PRF), also known as "blind feedback", simulates the RF scenario without having known relevant documents [Sparck Jones et al., 2000, Lavrenko and Croft, 2001]. Instead, the system *assumes* documents it ranks highly (i.e. predicts to be relevant) are indeed relevant and uses them as examples to improve and expand its interpretation of the original query. As such, PRF can be considered as a form of self-training or bootstrapping, and as with any such technique, it is important to model and propagate system uncertainty to achieve effective performance. With PRF in particular, we must carefully balance additional information from uncertain feedback against our limited but certain input query. While information from feedback documents can be incredibly effective in better inferring the latent information need, use of feedback can also result in harmful "concept drift" away from the true subject of interest. For example, term-based methods intending to reinforce key query terms and add related terms can also accidentally pull-in unrelated terms or lose emphasis on the most important terms. Consequently, feedback has been often seen to boost recall at some cost to precision. PRF effectiveness is clearly influenced by how accurately the system identifies documents to use for feedback and estimates their probability of relevance. While PRF can be iterated, multiple iterations usually hurt performance and so a single PRF iteration is most typical.

2.5.1 Relevance Feedback

Given a query, query-likelihood retrieval (Equation 2.4.17) infers relevance on the basis of similarity between (our estimates of) query and document models, Θ^Q and Θ^D. While discussion thus far has focused on document ranking for a given query, let us now consider the other direction of query formulation. Given a set of relevant documents $\mathcal{R}$ that match a user's information need, the optimal query model $\Theta^Q_\star$ under Equation 2.4.17 will exhibit greater similarity to $\mathcal{R}$'s latent document models $\forall_{D \in \mathcal{R}} \Theta^D$ than those of other documents. This suggests that given partial knowledge of $\mathcal{R}$ in the form of $|\mathcal{F}|$ feedback documents where $\mathcal{F} \subseteq \mathcal{R}$, Θ^Q might be estimated on the basis of similarity to $\mathcal{F}$. For example, a simple idea would be to estimate Θ^Q as the average document model over the

set of positive (i.e. relevant) feedback documents:

$$\widehat{\Theta^F} = \frac{1}{|\mathcal{F}|} \sum_{D \in \mathcal{F}} \Theta^D \tag{2.5.1}$$

While the classic Rocchio method [Rocchio et al., 1971] also incorporates negative feedback (γ term):

$$\vec{q_r} = \alpha\ \vec{q_0} + \beta \frac{1}{N_r} \sum_i^{N_r} \vec{d_i} - \gamma \frac{1}{N_{\bar{r}}} \sum_i^{N_{\bar{r}}} \vec{d_i} \tag{2.5.2}$$

negative feedback has typically been found to be far less useful than positive feedback. Since retrieval time is typically proportional to the number of terms used, a common efficiency heuristic is to approximate Θ^F by its k_F most likely terms and re-normalize[5].

Although the approach in Equation 2.5.1 does provide broader lexical coverage of $\mathcal{R}$ than available in the original query string, it suffers from a different problem. Whereas Q tends to closely focus on the core information need, the average feedback document model may diverge from it since documents in $\mathcal{F}$ likely discuss many topics. Rocchio's $\alpha\ \vec{q_0}$ mixing term helps prevent such drift. The same technique can be applied with query-likelihood by inferring Θ^Q on the basis of both the original query and the feedback documents in the form of a linear mixture:

$$\Theta^{Q'} = (1 - \lambda_F)\, \Theta^Q + \lambda_F\, \Theta^F \tag{2.5.3}$$

Despite the simplicity of this approach, recent studies have shown it achieves accuracy comparable to more sophisticated strategies [Balog et al., 2008, Yi and Allan, 2008].

Combining equations (2.4.16), (2.4.17), and (2.5.3), we see that unigram feedback can be equivalently interpreted as a mixture of query models under the unigram ranking 2.4.16 or as a mixture of ranking functions:

$$\begin{aligned} P(Q|D) &\overset{rank}{=} log\, \Theta^D \cdot \Theta^{Q'} \\ &= log\, \Theta^D \cdot [(1 - \lambda_F)\, \Theta^Q + \lambda_F\, \Theta^F] \\ &= (1 - \lambda_F)[log\, \Theta^D \cdot \Theta^Q] + \lambda_F [log\, \Theta^D \cdot \Theta^F] \\ &\overset{rank}{=} (1 - \lambda_F)\, \mathcal{D}(\Theta^Q || \Theta^D) + \lambda_F\, \mathcal{D}(\Theta^F || \Theta^D) \end{aligned}$$

However, as one moves away from unigram modeling to another retrieval model like the MRF (§2.6), we will see that this dual interpretation is no longer applicable.

2.5.2 Pseudo-relevance Feedback

PRF [Lavrenko and Croft, 2001] is quite similar to RF except that now we must factor in our uncertainty regarding each feedback document's relevance to the query. While our original setup in Equation 2.5.1 made a simplifying assumption that all feedback documents were equally relevant,

[5] Since Equation 2.4.17 is a linear model, ranking is invariant under any scaling of the weight vector and so normalization does not affect ranking. However, if we wish to later use Θ^F in some mixture model, choice of k_F will have a side-effect on mixture weight unless normalization is performed.

this estimate can be improved by accounting for varying degree of relevance across the feedback set. The straightforward way to accomplish this is to generalize from the simple average of Equation 2.5.1 to instead compute an expectation respecting some arbitrary estimate $p(D|Q)$ of feedback document relevance with respect to the query Q:

$$\Theta^P = E_{D \sim p(D|Q)}[\Theta^D] = \sum_{D \in C} p(D|Q)\ \Theta^D \tag{2.5.4}$$

where C denotes the document collection.

As with RF, a common efficiency heuristic is to approximate Θ^P by its k_P most likely terms and re-normalize. The original estimate of Θ^Q is also typically mixed with the Θ^P, similar to what was done with explicit feedback (Equation 2.5.3).

2.6 The Markov Random Field Model

Metzler and Croft's Markov random field (MRF) approach models a joint distribution $P_\Lambda(Q, D)$ over queries Q and documents D [Metzler and Croft, 2005]. It is constructed from a graph G consisting of a document node and nodes for each query term. Nodes in the graph represent random variables and edges define the independence semantics between the variables. In particular, a random variable in the graph is independent of its non-neighbors given observed values for its neighbors. Therefore, different edge configurations impose different independence assumptions. The joint distribution over the random variables in G is defined by:

$$P_\Lambda(Q, D) = \frac{1}{Z_\Lambda} \prod_{c \in C(G)} \psi(c; \Lambda) \tag{2.6.1}$$

where $C(G)$ is the set of cliques in G, each $\psi(\cdot; \Lambda)$ is a non-negative potential function over clique configurations parameterized by Λ, and $Z_\Lambda = \sum_{Q,D} \prod_{c \in C(G)} \psi(c; \Lambda)$ computes the partition function. For document ranking, we can skip the expensive computation of Z_Λ and simply score each document D by its unnormalized joint probability with Q under the MRF. If we define our potential functions as $\psi(c; \Lambda) = exp[\lambda_c f(c)]$, where $f(c)$ is some real-valued feature function over clique values and λ_c is that feature function's assigned weight, we can compute the posterior $P_\Lambda(D|Q)$ as

$$P_\Lambda(D|Q) = \frac{P_\Lambda(Q, D)}{P_\Lambda(Q)} \overset{rank}{=} \sum_{c \in C(G)} log\ \psi(c; \Lambda) = \sum_{c \in C(G)} \lambda_c f(c) \tag{2.6.2}$$

The graph G can be constructed in various ways depending on various possible assumptions regarding independence between terms. In the case of *full independence*, query term nodes share an edge with the document only. With *sequential dependence*, adjacent terms in the query share an additional edge in G. Finally, assuming *full dependence* constructs an edge between each pair of query term nodes. The choice of graph structure determines the set of cliques present in G and thereby the set of features used in ranking.

2.6.1 The Features

All of the potential functions used in the MRF can be expressed in the following generic form:

$$log\,\psi_i(c;\Lambda) = \lambda_i f_i(c) = \lambda_i log\left[(1-\alpha_i^D)\,\frac{S_i(c)}{|D|} + \alpha_i^D \frac{S_i(c)}{|C|}\right] \tag{2.6.3}$$

where $S_i(c)$ denotes a given statistic computed for the given clique c, $|D|$ and $|C|$ indicate respective token counts of the document and entire collection (statistics other than term frequency are only approximately normalized), and $\alpha_i^D = \frac{\mu_i}{\mu_i+|D|}$, where μ_i denotes a smoothing hyper-parameter specific to the potential function $\psi_i(c;\Lambda)$ [Zhai and Lafferty, 2004]. Note that use of term frequency as the statistic S_i computes the standard Dirichlet-smoothed unigram (2.4.14).

Potential functions are primarily distinguished by the particular statistic S_i they employ. As mentioned earlier (§1.1), the MRF model exploits three classes of lexical features: individual terms, contiguous phrases, and proximity. Each of these corresponds to a distinct statistic S_i: term frequency, phrase frequency (i.e. "ordered" Indri `#1` operator), and frequency of a set of terms within some parameter N-sized window (i.e. "unordered" Indri `#uwN` operator). The latter two multi-term statistics' corresponding potential functions are applicable when some form of dependency is assumed between query terms in the graph structure. In particular, the phrasal potential function is only applied to cliques connecting contiguous query terms, whereas the proximity potential function is applied to all multi-term cliques, contiguous and non-contiguous alike. This means each pair of contiguous query terms generates a clique c whose potential function is defined by the product $\psi_o(c)\psi_u(c)$ of ordered and unordered potential functions.

Using these three classes of potential functions, the MRF can be expressed as a three component mixture model computed over term, phrase, and proximity feature classes:

$$\sum_{c\in C(G)} \lambda_c f(c) \;=\; \sum_{c\in T} \lambda_T f_T(c) \;+\; \sum_{c\in O} \lambda_O f_O(c) \;+\; \sum_{c\in O\cup U} \lambda_U f_U(c) \tag{2.6.4}$$

Each class effectively computes its own ranking function which is then mixed with that of the other classes. Ch. 4 shows how assumptions underlying estimation of each class can be relaxed to improve search accuracy.

2.6.2 Pseudo-relevance Feedback

Recall our basic PRF equation (2.5.4) computing an expectation over documents. Whereas we were able to skip normalization in (2.6.2) since we were only using the model for ranking, to compute the PRF expectation we need a normalized probability distribution. However, we need not compute the full partition function to normalize $P_\Lambda(Q,D)$ over the entire document collection unless we want to use the entire collection for feedback. Besides the large computational cost this would incur, there is diminishing return and increasing harm from query drift as we start sifting through lower ranks. Instead, we can simply normalize with respect to the set of PRF documents $\mathcal{P}$ only:

$$P_\Lambda^N(D|Q) = \frac{P_\Lambda(Q,D)}{\sum_{D\in\mathcal{P}} P_\Lambda(Q,D)} \tag{2.6.5}$$

Collection	Avg. Term Count	Std. Deviation
CACM	10.80	6.43
CISI	28.29	19.49
CRAN	9.17	3.19
INSPEC	15.63	8.66
MED	10.10	6.03
NPL	7.16	2.36

Table 2.1: Query length statistics for several classic document collections [Salton and Buckley, 1987]. NPL queries were considered "very short" while CISI and INSPEC queries were considered "long".

The expected PRF document model can then be computed by (2.5.4) as with unigram PRF.

While PRF could potentially be used to better estimate all three classes of MRF features, previous work has shown little benefit from applying this technique to estimation of adjacency and proximity classes (f_O and f_U, respectively) [Metzler and Croft, 2007a]. We are not aware of any work attempting to better estimate these classes via explicit RF either.

2.7 Verbose Queries

This section provides a brief and highly selective history of search using verbose queries. We begin by discussing data-specific issues: the evolving notion of queries over time, different taxonomies employed, characteristic statistics, associated document collections, etc. Following this, we summarize previous work: methods and corresponding results obtained.

2.7.1 Data

Life Before TREC (1992)

As an ultra-brief snapshot of queries used prior to TREC, query length statistics for several pre-TREC (and pre-Web) datasets given in [Salton and Buckley, 1987] are summarized in Table 2.1. Relative to the sort of terse, keyword queries more typical of Web search today [Broder, 2002], queries used with these collections were generally much longer. While the NPL queries (and documents) were relatively short in comparison to the other collections, the authors suggest this may be due to indexing vocabulary in NPL being carefully selected rather than all terms having been indexed[6]. An interesting parallel to consider is the effects of index term selection vs. user selection of (keyword) query terms, as both tend to lead to shorter queries with different distributional properties than one typically finds in more natural language. We discuss this issue further in §2.7.2.

[6] Only the numerical query and document vectors were publicly available; original texts were not [Salton and Buckley, 1987].

TREC Topic 331	
`title`	`World Bank Criticism`
`description`	`What criticisms have been made of World Bank policies, activities or personnel?`
`narrative`	This query is looking for any instances where the World Bank has been accused of things like not being responsive to the unique problems of individual countries, of being too strict in its policies, of pursuing agendas that are biased because of their benefits to western countries, of being no longer useful or practical, of its personnel being difficult to work with, etc.

Table 2.2: An example TREC topic.

TREC	Year	Topics	**Min**	**Max**	**Mean**
1	1992	51-100	5	41	17.9
2	1993	101-150	6	41	18.7
3	1994	151-200	9	42	22.3
4	1995	201-250	8	33	16.3
5	1996	251-300	6	40	15.7
6	1997	301-350	5	62	20.4
7	1998	351-400	5	34	14.3
8	1999	401-450	5	32	13.8

Table 2.3: Statistics for length in tokens of the `description` field of TREC 1-8 topics.

The Text Retrieval Conference (TREC)

Since 1992, significant cross-system evaluation has been performed at annual Text Retrieval Conference (TREC) evaluations[7]. As part of these evaluations, the National Institute of Standards (NIST) has been heavily involved in defining query topics for search and performing relevance annotations in the Cranfield tradition so that systems could be quantitatively evaluated. TREC has had an incredibly influential impact on the field of IR in general, and this section focuses on the evolving notion of topics and queries across TREC campaigns. Material presented in this section brings together information dispersed across the annual TREC overview reports available online; more specific citations are given where appropriate.

Original ad hoc topics from TREC 1-2 consisted of four fields providing reflecting differences in length, verbosity, detail, format, etc.: `title`, `description`, `narrative`, and `concepts`. The `concepts` field was dropped after TREC-2 but has still influenced derivative work (cf. [Zhai and Lafferty, 2004, Fang et al., 2004, Mei et al., 2007]). TREC-3 fields were generally shorter in comparison to previous years, but participants still felt they were too long compared with "what users normally submit to operational retrieval systems". Consequently, TREC-4 defined the topic simply by a one sentence description of the information need. However, the omission of a `narrative` field

[7] http://trec.nist.gov

Collection	Documents	Unique Terms	Tokens	Topic IDs
Robust04	528,155	643,239	276,914,688	301-450, 601-700
W10g	1,692,096	5,368,332	1,066,462,974	451-550
GOV2	25,205,179	39,294,014	27,047,041,080	701-850

Table 2.4: Example TREC Collections and associated Topic IDs. Topics 672 and 703 have no relevant documents and therefore do not impact evaluation.

for guiding assessment and interpretation of topics was seen as a significant loss. Consequently, the three-field `title`, `description`, and `narrative` format was restored in TREC-5 and has continued since. An example topic, 331, is shown in Table 2.2.

In terms of analyzing search accuracy as a function of topic field(s) used, little distinction was made in TREC 1-3. As mentioned above, TREC-4 topics consisted of a `description`-like field only. Over the next several years, a four-way taxonomy of queries emerged in analyzing the effect of different query types on search accuracy [Sparck Jones, 1999]:

1. very short: `title` field (TREC-6)
2. short: `description` field (TREC-5)
3. medium: `title` and `description` concatenated (TREC-7)
4. long: `title`, `description`, and `narrative` concatenated (TREC-5)

Nevertheless, TREC-8 referred to `title` and `description` concatenated as "short", so it seems the notion of query categories remained somewhat fluid.

With topics 301-450 (TREC 6-8), the `title` field was specifically designed to allow experiments with very short queries consisting of at most three words. The `description` field was intended to be a one sentence version of the query, as in TREC-4, but in practice this field may consist of multiple sentences (e.g. topics 342-347). An important change with TREC-7 topics was that the `description` field was intentionally written to use all terms found in the `title` field to avoid confounding effects of verbosity with missing terms. While this also largely holds for TREC-8 topics (except 413), subsequent topics developed often do not use all `title` field vocabulary in the `description` field, complicating comparison between exclusive use of either field.

With regard to document collections, TREC 6-8 all used the same collection, later re-used in the TREC 2003-04 Robust tracks. An additional 100 topics, 601-700, were also created for the 2003-04 Robust tracks. We refer to this collection and the 250 topics as Robust04 (Table 2.4).

The same topic formulation process was also used to create topics 701-850 for the Terabyte track (TREC 2004-2006) using the GOV2 collection (Table 2.4). This document collection corresponds to a crawl of the .gov portion of the Web conducted early in 2004, and topics were intended to reflect informational Web queries (as opposed to navigational or transactional [Broder, 2002]).

Topics 451-550 (TREC 9-10, 2000-2001) were created for the Web track's W10g collection (Table 2.4) via a different process than the other topics discussed thus far. Participants wanted topics to

strongly resemble queries typically employed in Web search. Rather than having assessors formulate the ideas for topics, NIST instead obtained a log of real queries submitted to commercial search engines (Excite queries from Dec 20, 1999 for TREC-9, and from MSNSearch in TREC-10). Query strings selected from these logs were then used verbatim to create the `title` fields of the new topics. NIST assessors retrofitted `description` and `narrative` fields around their interpretation of intent underlying these queries, resolving any ambiguity. Spelling was left uncorrected in topics 451-500 but was corrected for topics 501-550. No attempt was made to correct grammar of the `title` queries, though other topic fields were intended to reflect grammatical (American) English.

Queries Based on TREC Topics

Zhai and Lafferty [Zhai and Lafferty, 2004] differentiated two types of queries, "title" and "long", in order to study the impact of different smoothing strategies for different types of queries. Their definition of these two categories was equivalent to the TREC-6 categories of "very short" and "long", using the `title` field and the `title`, `description`, and `narrative` concatenated, respectively. Unfortunately, they subsequently realized a confounding factor in this division: long queries were both longer and reflected a different distribution of common vs. topical vocabulary. Consequently, they further refined their distinction among queries to four categories in follow-on work [Zhai and Lafferty, 2002]. Once more TREC data was used in experiments, but their four-way taxonomy was different than that used in TREC-7:

1. short keyword (`sk`): `title` field
2. short verbose (`sv`): `description` field
3. long keyword (`lk`): `concept` field found in early TREC topics
4. long verbose (`lv`): `title`, `description`, and `narrative` concatenated (earlier "long" category)

While short and long keyword queries both consisted of only strong content terms, short and long verbose queries reflected a term distribution more characteristic of natural language. Thus distinctions between query verbosity vs. length were teased apart. In addition to enabling the particular study in [Zhai and Lafferty, 2002], this query taxonomy has continued to prove useful to Zhai and his students in analyzing the behavior of various retrieval methods (cf. [Fang et al., 2004, Mei et al., 2007]).

2.7.2 Previous Work

This section briefly reviews methods and results of previous work for search using verbose queries.

Salton and Buckley, 1987 [Salton and Buckley, 1987]

In work preceding the TREC program, Salton and Buckley provided insights, experimental results on several document collections (Table 2.1), and recommendations for query term weighting as a function of query length. Several combinable document and query term weighting strategies were

described which they identified via a pair of triples of the form $ddd{\cdot}qqq$. The first triple characterized document term weighting, and the latter, that of the query. Each triple expressed options for term frequency (TF), inverse document frequency (IDF), and length normalization (LN), respectively:

1. TF: binary weight (b), normal TF (t), normalized TF (n)
2. IDF: none (x), normal (f), and probabilistic (p)
3. LN: absent (x) or present as cosine (c)

For weighting query terms, all choices for LN are rank-equivalent and simply scale document scores. Simple boolean weighting (is the term present or absent regardless of frequency) is achieved by bxx while txx weights terms by simple relative frequency (equivalent to language modeling §2.4.3 term-wise query-generation or maximum-likelihood estimation of Θ^Q). Either choice is equivalent if all query terms occur exactly once.

Two methods were seen to generally perform best and consistently with one another across document collections (except for NPL): "fully weighted" $tfc{\cdot}nfx$ and "probabilistic weight" $nxx{\cdot}bpx$. It is interesting to note that the former applies IDF (f) weighting to query terms as well as document terms, effectively squaring IDF; the authors noted this "enhanced query weighting" was particularly effective. With the NPL's very short queries and their minimal length deviation (as well as its short documents), however, boolean (b) weighting of query terms was seen to be most effective. In contrast, non-boolean weighting of query terms was seen to be essential with the long queries of CISI and INSPEC. Given the similar effect of controlled vocabularies (§2.3) and keyword search in tending to yield shorter, more carefully chosen terms, experience here with weighting query terms in NPL vs. other document collections may also inform term weighting for keyword vs. verbose queries.

The authors concluded with several general recommendations for weighting query terms. Regarding TF, "for short query vectors, each term is important; enhanced query term weights (n) are thus preferred... long query vectors require a greater discrimination among query terms based on term occurrence frequencies (t)... ." As for IDF, they reported the most effective methods used IDF weighting (f).

TREC

As in the previous section on TREC data (§2.7.1), material presented in this section brings together information dispersed across the annual TREC and task-specific overview reports available online from NIST[8]; more specific citations are given as needed.

Search based on the `description` field only of topics was evaluated in TREC 5-6 the Robust track (2003-2005), and effectively in TREC 4 where the entire topic was `description`-like. The more common trend at TREC has been to differentiate between different levels of query verbosity by concatenating shorter fields with longer ones (e.g. `title`, `title` + `description`, and `title` + `description` + `narrative`). While this certainly does test a different verbosity condition, the usage model seems somewhat odd: while we might imagine a user requesting information via a natural

[8] http://trec.nist.gov

language question or sentence, corresponding to the `description`-field, it seems much less likely they would simultaneously express their query at multiple levels of detail, e.g. `title` + `description`. This is of further importance because these two conditions, `description` vs. `description` + `title` behave rather differently: the latter benefits from having keywords from the `title` emphasized through repetition and so solves a less difficult and interesting problem in trying to effectively cope with verbose natural language.

Difference in performance between `title` only and `description` only fields was observed and much discussed in TREC-6, but discussion largely centered around important terms from a topic's `title` field missing from its `description` field, causing a confounding effect between verbosity and important terms being missing. While topics were largely fixed for TREC 7-8 to ensure all `title` field terms also appeared in the `description` field, there appears to have been relatively little re-examination of the difference in performance between alternate use of each field now that the confounding effect had been removed. It may be that query expansion techniques had become sufficiently effective to largely close the gap between `title` vs. `description` runs [Sparck Jones, 2000]; this needs to be further investigated.

A list of strong performing systems in TREC 5-8 which ranked documents using only the `description` field of topics can be found in [Sparck Jones, 1999, 2000] (along with the approximate precision-at-30 search accuracy each achieved). It should also be noted that actual document ranking of participating systems over all TREC evaluations can also be officially obtained from NIST, allowing more thorough analysis and comparison between various systems that have been employed.

As in TREC 5-7, Robust 2003-2004 tracks again required participants to rank documents using on the `description` field of topics. TREC 6-8 topics were also reused. A goal of the task was to investigate system performance on difficult topics, and while the problem of TREC-6 `description` fields missing important terms was already known, no mention is made of it in the respective track overviews. However, per-topic median average precision scores graphed in the 2003 track overview suggest the difficult TREC-6 topics used did not perform worse on average than those from TREC 7-8. The most meaningful comparison can likely be made to each year's blind evaluation topics: 601-650 for Robust 2003, and 651-700 for Robust 2004. In addition, a particular set of 50 hard topics evaluated each year are identified in the 2003 track overview; results here are something of an upper-bound since systems could tune on this set of topics during development. Finally, the Robust04 track also evaluated over all 250 topics (Table 2.4), mixing the 50 blind topics with the 200 known topics. Testing conditions aside, results on this complete set of topics can be directly compared to other results on the Robust04 collection presented in this work as well as that of cited previous work.

While there was also a Robust track in 2005 that required participants to submit both `title` and `description`-field submissions, search was performed on the AQUAINT document collection rather the Robust04 collection. The track overview for this year presents one of the more interesting discussions of comparative search accuracy with `title` vs. `description`-fields. In particular, while

description queries were seen to perform significantly worse, analysis showed this to be a flaw in how NIST performed its pooled assessments. Given the much larger collection size, there were many more relevant documents for each topic yet pool sizes did not grow proportionately. The title field-only runs effectively found enough relevant documents to fill up the pools such that relatively fewer documents found by description field runs to be assessed, thus biasing the evaluation. As such, future use of this track's assessments was advised against without additional assessment to correct for this bias. Consequently, we do not evaluate on this collection. The track overview also remarked that "...title-only runs are more effective than description-only runs for the AQUAINT collection, while the opposite is true for the [TREC 6-8/Robust04] collection". At present we are unaware of the particular analysis or evidence providing the basis for that comment; comparative results shown in the Robust04 track overview suggest comparable accuracy of strong-performing systems (using query expansion) for title only and description only runs.

In terms of comparing search accuracy for difficult topics considered in the Robust tracks, comparison here is somewhat more involved. Mean-average precision (MAP) is easily compared: it was reported by track and is computed by trec_eval[9] for evaluating new systems. However, a poorly performing topic would have to change dramatically to impact MAP, and so it is not ideal for this sort of evaluation. Robust tracks instead reported two non-standard metrics: % of topics for which precision-at-10 is zero and area under the MAP curve for the worst quarter topics. These two metrics were superseded by geometric MAP (gMAP), described in the Robust 2004 track overview but not reported for participating systems. Although gMAP was integrated into trec_eval for Robust05, the other metrics were abandoned. While the metrics are not difficult to implement, a special evaluation script for computing them is also publicly available[10] and ensures consistency with how official results were computed. The track overviews also report these metrics were less stable than gMAP for the same number of topics. To compare accuracy on the basis of gMAP, official document rankings for the two tracks must be obtained from NIST since gMAP was not reported.

The Web track which used the W10g corpus (TREC-9, TREC 2001) and performed blind evaluation on topics 451-550 (Table 2.4) focused on title field only runs and report only those in the track overview. Again, one needs to obtain official rankings from NIST or read through the individual participant papers or results in the appendix of each year's proceedings to obtain additional detail regarding other topic fields used and resulting accuracies achieved.

TREC Terabyte tracks used the GOV2 corpus (TREC 2004-06) and performed blind evaluation on topics 701-850 in groups of 50 over the three years the track was run. As with the earlier Web track, participation in automatic search required title field only runs, and track overview papers summarize those results. As an exception the 2004 Terabyte track overview lists submitted runs, showing only one team submitted description field only runs that year. As with the Web track, further investigation is needed to determine what use of the description was made other years and the resulting effectiveness.

[9] http://trec.nist.gov/trec_eval

[10] http://trec.nist.gov/data/robust/robust2004_eval.pl

The Query track (TREC 7-9) produced 2000 variant queries for TREC-1 topics (51-100) in order to study the effect of query formulation upon retrieval accuracy. By generating many queries for the same topic, the hope was to better understand the nature of topic difficulty by disentangling *how* an information need is expressed vs. *what* the information need is. Two primary types of queries were generated: "(very) short" queries of 2-4 words and "sentence" queries of 1-2 sentences (a third type we will not discuss created queries without reference to the original topics, which led to topic drift). The TREC-8 track overview noted that short queries performed noticeably better than sentence queries. A confounding factor, however, was that almost all the sentence queries were generated by students whereas half the short queries were generated by experts. Re-examination without this confounding factor was recommended but was not further discussed.

The TREC 2005 Question Answering (QA) track included a document ranking task to investigate whether some document retrieval techniques are better than others in support of QA. In other words, documents were ranked in response to question-type queries, and systems searched for answers using these ranked documents (although some QA system architectures did not produce an explicit list of document rankings as an initial step). Only a weak correlation was found between document retrieval accuracy and system ability to answer factoid-type questions (i.e. questions like "Who shot Abraham Lincoln?" seeking a simple fact in response). Beyond enabling the specific study performed in the track, the pool of document rankings produced was intended to serve subsequent research by the community in the interaction between IR and QA accuracies.

The 2003 Reliable Information Access (RIA) Workshop [Buckley and Harman, 2004]

The six-week RIA Workshop investigated contributions from retrieval system variability factors and topic variability factors in order to better understand overall variability in search accuracy. As part of this workshop, massive failure analysis was performed using document rankings generated by six different retrieval systems. In particular, the workshop was very interested in the intersection of IR and QA: initial search for documents in response to question-type queries, and the value of IR as a backup option in case focused retrieval [Kamps et al., 2008] should fail. Consequently, the workshop studied rankings with verbose queries for a 45 topic subset of TREC 6-8 (topics 301-450); 26 of these topics overlapped the set of "hard" topics evaluated in 2003-2004 Robust tracks. It should be noted that the `description` fields were not used verbatim, but rather a standard set of patterns was defined to be filtered out of all queries (i.e. phrasal stopping); the idea was to filter out highly stylized language such as "a relevant document must identify..." which users would be unlikely to employ in formulating queries. In total, 28 people from 12 organizations were involved in the effort, and from 11-40 person-hours was spent analyzing each topic. In addition to findings summarized in the workshop report, a website was also developed to provide access to the detailed topic-specific analysis performed[11]. Another product of the workshop Chris Buckley's ten-way taxonomy of queries which distinguished between the anticipated sophistication in natural language understanding that would be needed to improve performance on a given query were its category in the taxonomy known. For at

[11] http://ir.nist.gov/ria

least half the categories, current technology was seen to be sufficient to significantly improve results. This recognition, along with further analysis of this taxonomy mentioned in §1.3.1, inspired our work that better term weighting could significantly improve search accuracy with verbose queries.

Question Answering (QA)

As suggested in the previous two sections, there is a close connection between IR and QA; natural language questions represent one form in which queries may be expressed, focused retrieval [Kamps et al., 2008] ranging from a paragraph to a few words can allow users to find the desired information more quickly than returning an entire document, and IR techniques are often employed as part of QA systems. The relationship between IR and QA was studied at a SIGIR workshop in 2004 [Gaizauskas et al., 2004], the TREC 2005 QA track contained a specific document ranking task (§6.2), and Lin presents an interesting look at the interaction for more complex relationship questions rather than factoid questions [Lin, 2006]. Lin reports that for factoid questions, existing work on question analysis has investigated strategies for identifying important query terms, which is another way to say term weighting strategies for question-type queries. Consequently, we are interested in further exploring this line of research to see if its findings can generally applied to improving document ranking for verbose queries.

Two-Stage Smoothing [Zhai and Lafferty, 2002, 2004]

Earlier we discussed Zhai and Lafferty's evolving query taxonomy (§2.7.1). In this section we discuss their findings and methodology they developed.

In their initial study, Zhai and Lafferty observed strong interactions between the type of smoothing employed and type of queries used for retrieval [Zhai and Lafferty, 2004]. Unfortunately, it was unclear from their initial study whether the high sensitivity observed on longer queries was the result of those queries' more frequent use of common terms or simply their length. In follow-on work [Zhai and Lafferty, 2002], they posited it was the former, and to test this hypothesis, they defined four types of queries mentioned earlier. It is worthy of mention that no stopwords were removed in their study, simplifying analysis of their methods and results. They found that the two types of keyword queries behaved similarly with respect to smoothing, as did the two types of verbose queries. In particular, they observed that it was the verbose queries that were much more sensitive to smoothing, supporting their hypothesis that verbosity rather than length was the source of smoothing sensitivity observed in their earlier work. They also observed a consistent order of performance among the four types of queries, with the `description` queries always under-performing the `title` queries.

Zhai and Lafferty's analysis of this effect was that smoothing plays two distinct roles in query-likelihood retrieval: to better estimate document models and to "explain away" common and non-discriminative words in the query so that document ranking would be primarily a function of topical terms rather than terms arising from (verbose) natural language. To model this effect, they proposed a generation process where by query terms arise from a mixture of two multinomials: the topical

document model Θ^D and a query background model $P(\cdot|\mathcal{U})$:

$$p(Q = q_{i:m}|\Theta^D, \lambda, \mathcal{U}) = \prod_{i=1}^{m}(1-\lambda)p(q_i|\Theta^D) + \lambda p(q_i|\mathcal{U}) \tag{2.7.1}$$

They adopt the ML collection unigram Θ^C for $p(\cdot|\mathcal{U})$ and estimate Θ^D by Dirichlet smoothing. Moreover, they achieve a parameter-free model in the following way. First, we assume each document D arises from a mixture between a latent document unigram Θ^D and a prior unigram, with the mixture parameter set by Dirichlet hyper-parameter μ. Next, we can compute the likelihood of the observed collection C under the assumption that these latent unigrams correspond exactly to observed relative frequency statistics (i.e. ML) for D and C:

$$\begin{aligned}
\log p(C|\mu) &= \sum_{d \in C} p(d|C, \mu) \\
&= \sum_{d \in C} \sum_{w \in V} c(w, d) \log p(w|d, C, \mu) \\
&= \sum_{d \in C} \sum_{w \in V} c(w, d) \log \left(\frac{c(w, d) + \mu\, p(w|C)}{|d| + \mu} \right)
\end{aligned} \tag{2.7.2}$$

Zhai and Lafferty set μ by maximizing a "leave-one-out" variant of this likelihood:

$$\begin{aligned}
\log p_{-1}(C|\mu) &= \sum_{d \in C} \sum_{w \in V} c(w, d) \log p_{-1}(w|d, C, \mu) \\
&= \sum_{d \in C} \sum_{w \in V} c(w, d) \log \left(\frac{c(w, d) - 1 + \mu\, p(w|C)}{|d| - 1 + \mu} \right)
\end{aligned} \tag{2.7.3}$$

Next, expectation maximization (EM) is run to learn a λ for each query which maximizes its likelihood given the Dirichlet-smoothed document unigrams and the value of μ learned above.

Query Reduction [Kumaran and Allan, 2007, 2008, Kumaran and Carvalho, 2009]

While query expansion methods try to improve search accuracy via augmenting a user's query with additional terms, Kumaran et al. have explored methods of "query reduction" for removing terms from the input query, rewriting a verbose query as a smaller subset of the original terms to effectively transform it into a keyword query. The idea of query reduction is to remove terms from the input query that risk obscuring the core information need. As such, query reduction can be seen as a generalized form of stopword removal (§2.3) that dynamically chooses terms to stop.

Query reduction can also be viewed as a restricted form of term weight estimation in which terms are assigned binary weights: terms assigned zero weight are effectively stopped, and remaining terms are assigned equal weight, matching the uniform distribution of weights estimated by standard maximum-likelihood (ML) estimation (§2.4.3). Why remove terms when they can be more flexibly weighted? If we want to leverage user interaction, query reduction suggests a fairly simple and intuitive model of interaction [Kumaran and Allan, 2007, 2008]. While one could envisage a user interface allowing users to assign weights to terms[12], choosing between alternative query candidates

[12] http://searchcloud.net

with varying term weights could be cognitively more challenging than simply choosing between subsets of terms.

While large search accuracy gains were originally shown via user interaction, recently an effective fully-automated system been demonstrated [Kumaran and Carvalho, 2009]. Evaluation on verbose queries is performed on a subset of topics with TREC 1-3 documents (topics 51-200) and Robust04 (Table 2.4). Improvements over baseline Dirichlet smoothing (§2.4.3) are shown, but comparison with other systems is complicated by the unique subset of topics used in evaluation. Although the set of topics used was not reported, it likely can be obtained directly from the authors to allow such direct comparison.

Key Concepts [Bendersky and Croft, 2008]

Key Concepts posits a core concept at the heart of each verbose query and tries to automatically detect it. In particular, an automatic chunker is run on each query to identify all base noun phrases (NPs). NPs are then manually annotated to identify a single key concept for each query. A supervised classifier is trained using various features to predict which detected NP in each verbose query is the key concept. Analysis found that most verbose queries considered could be faithfully represented by one or two such NPs, and evaluation showed the classifier is often quite effective in finding them.

Following this, the classifier was applied to weight terms as follows. Each detected NP is weighted by classifier confidence that it is the given query's key concept. Next, these NP weights are propagated to weight individual terms: all terms in a given NP are given an equal portion of the NP's total predicted weight. Because the model weights NPs only, query terms outside of NPs would receive zero weight without smoothing, potentially reducing model robustness. To address this, Θ^Q is instead estimated as a two-component mixture model combining the model's predicted term weights with the uniform maximum-likelihood (ML) estimate (§2.4.3). A single mixture weight determined by cross-validated tuning is used across collections and topics. Document ranking accuracy was evaluated using the collections and topics shown in Table 2.4. Effectiveness of Key Concepts' method of term weighting is compared to that of other methods in the following section. Further discussion of the Key Concepts approach to term weighting is presented in §3.4.

Comparison of Previous Work

Table 2.5 summarizes search accuracies for verbose queries achieved by existing methods. Some results are copied from published work [Zhai and Lafferty, 2002, Mei et al., 2007] and TREC proceedings while others we directly evaluated ourselves [Metzler and Croft, 2005, Smucker and Allan, 2006, Bendersky and Croft, 2008]. Results with verbose queries were first reported for Dirichlet smoothing in [Zhai and Lafferty, 2002] and for the MRF model in [Metzler, 2007, Bendersky and Croft, 2008]. Indri queries produced by Key Concepts (§2.7.2) were provided to us by the authors. Smucker and Allan's model incorporates an inverse collection frequency (ICF) factor to capture and IDF-like effect in weighting query terms [Smucker and Allan, 2006].

Results with verbose queries on TREC-4 topics reported in [Gao et al., 2004, Na et al., 2008] are not shown. In a non-TREC publication [Fan et al., 2004], Fan et al. describe evaluation of verbose queries per their participation in the Robust03 track (i.e. their "`VTDokrcgpa5`" submission). Hoenkamp et al. [Hoenkamp et al., 2009] describe the "epi-HAL" technique based on query expansion using the Hyperspace Analog to Language (HAL) and evaluate verbose queries for TREC-2 topics on the AP88-89 document collection and for Robust04 topics and documents. We compare to their latter results in §5.2.

Table 2.6 compares search accuracy of verbose queries of published methods vs. official results of the Robust track at TREC 2004. One of the top performing systems, `pircRB04d4` employed phrases, PRF [Lavrenko and Croft, 2001], and query expansion via the Web [Kwok et al., 2005]. Such Web expansion was highlighted in the track overview as a key component in the track's competitive systems, and such Web expansion is typically not employed in published IR work. It is interesting to note, however, that even without Web expansion the model was quite strong (and in many cases actually better performing without the Web expansion). As such, it seems that inclusion of query expansion via the Web does not by itself explain the relative strength of this system vs. other published results.

Official TREC results for verbose queries typically exceed search accuracies reported in other published research. Regarding the one exception shown here, `INQ602`, while it was the strongest `description` field-only run the given year, it was generally not competitive with other participating groups, as seen by the wide disparity vs. the `pir9At0 title` field results that same year. The apparent superiority of competitive TREC systems is due in large part to their almost always applying query-expansion techniques like pseudo-relevance feedback (PRF, §2.5.2) while published research often does not. Because expansion techniques can typically be applied atop any non-feedback method, published research often decouples these two problems to separately investigate non-feedback methods vs. expansion techniques. To further generalize this point, published research often investigates particular aspects or subtasks of retrieval, comparing to other work studying the same given subtask, while TREC evaluations emphasize overall system accuracy. An interesting recent study comparing results of TREC evaluations over time (across TREC topic fields) provides a critical examination of the field's quantitative progress over the past fifteen years [Armstrong et al., 2009]. Its presentation also compared published results vs. those reported in TREC evaluations.

	System	Year	TREC-7 P@5	TREC-7 MAP	TREC-8 P@5	TREC-8 MAP
	`title` only					
TREC	Okapi `ok7as` (TREC-7)	1998	53.20	26.14		
	Queens `pir9At0` (TREC-8)	1999			51.60	30.63
	`description` only					
TREC	NEC `nectitechdes` (TREC-7)	1998	58.40	25.84		
	UMass `INQ602` (TREC-8)	1999			49.60	24.92
No PRF	Dirichlet Smoothing (§2.4.3)	2001	46.80	17.96	44.80	23.26
	Two-Stage Smoothing (§2.7.2)	2002	41.60	18.10	48.40	23.10
	MRF (§2.6)	2005	48.40	18.95	46.80	23.71
	Collection-ICF [Smucker and Allan]	2006	50.40	20.11	46.00	24.77
	Term Dependent Smoothing [Mei et al.]	2007	44.00	19.60	47.60	24.60
	Key Concepts (§2.7.2)	2008	48.00	20.21	46.00	23.64
With PRF	Dirichlet Smoothing (§2.4.3)	2001	46.80	23.12	52.00	27.11
	MRF (§2.6)	2005	50.80	23.85	53.20	28.34
	Collection-ICF [Smucker and Allan]	2006	49.20	24.79	50.80	28.09
	Key Concepts (§2.7.2)	2008	48.80	24.41	50.80	27.28

Table 2.5: Search accuracy in mean-average precision (MAP) and precision of top 5 ranks (P@5) for verbose queries (`description` field) on the Robust04 collection (Table 2.4) using topics from TREC-7 (351-400) and TREC-8 (401-450). While the competitive TREC systems employed query expansion techniques, the "without PRF" systems did not. To give a general sense for this difference, we produced results with PRF [Lavrenko and Croft, 2001] for several methods using Indri [Strohman et al., 2004] parameters shown in Table 2.6. While one of the parameters was tuned for the MRF, all of the methods stand to benefit from better tuning. Nonetheless, results clearly show that all benefit substantially from PRF in terms of resultant MAP accuracy. Official TREC results reflect blind evaluation; other testing conditions vary. Statistical significance is not reported.

ID	Old Topic Set				New Topic Set				Hard Topic Set				Combined Topic Set			
	MAP	P10	%no	area	MAP	P10	%no	area	MAP	P10	%no	area	MAP	P10	%no	area
t	.317	.505	5	.033	.401	.545	6	.089	.183	.374	12	.016	.333	.513	5	.038
d1					.3503		4	.0638	.1182		14	.0063	.2342		9	.0121
d2					.4044		4	.0839	.1524		30	.0049	.2784		17	.0125
d3	.315	.507	8	.023	.407	.547	2	.074	.162	.382	12	.013	.334	.515	7	.028
Without PRF																
1	.2318	.4025	11.5	.0098	.2993	.4673	4.1	.0426	.0988	.2560	20.0	.0054	.2451	.4153	10.0	.0118
2	.2413	.4200	10.5	.0122	.3180	.4735	4.1	.0540	.1096	.2920	14.0	.0064	.2564	.4305	9.2	.0149
3	.2482	.4085	13.5	.0084	.3058	.4653	4.1	.0378	.1092	.2700	20.0	.0055	.2595	.4197	11.6	.0101
4	.2456	.4015	13.5	.0078	.3141	.4776	4.1	.0360	.1057	.2440	22.0	.0022	.2591	.4165	11.6	.0100
With PRF [Lavrenko and Croft, 2001]																
1	.2660	.4315	16.5	.0065	.3770	.4939	6.1	.0678	.1157	.2820	26.0	.0019	.2879	.4438	14.5	.0089
2	.2792	.4485	16.5	.0072	.3897	.5082	6.1	.0786	.1309	.3060	24.0	.0024	.3009	.4602	14.5	.0104
3	.2799	.4380	15.5	.0069	.3725	.5000	8.2	.0567	.1240	.2860	24.0	.0039	.2981	.4502	14.1	.0088
4	.2749	.4240	17.0	.0046	.3694	.4857	12.2	.0344	.1145	.2620	30.0	.0011	.2935	.4361	16.1	.0060

Best TREC Robust04 `title` (t) and `description` (d3) runs
t. `pircRB04t3`
d1. `pirc` model without PRF or web expansion [Kwok et al., 2005]
d2. `pirc` model with PRF but without web expansion [Kwok et al., 2005]
d3. `pirc` model with PRF and web expansion (official `pircRB04d4` submission)

Other Methods
1. Dirichlet Smoothing [Lafferty and Zhai, 2001]
2. MRF (§2.6)
3. Collection-ICF [Smucker and Allan, 2006]
4. Key Concepts (§2.7.2)

PRF [Lavrenko and Croft, 2001] results were generated with the following Indri [Strohman et al., 2004] settings:

1. `fbDocs` = 10 (fixed)
2. `fbTerms` = 50 (fixed)
3. `fbMu` = 0 (default)
4. `fbOrigWeight` = 0.4 (tuned for MRF on topics 301-450)

Table 2.6: Comparison of published work vs. official results of the TREC 2004 Robust track (refer to Table 3 in the 2004 track overview). Evaluation is performed on the Robust04 document collection (Table 2.4) using four topic sets defined by the track: "old" (301-450, 601-650), "new" (651-700), "hard" (50 topics from 301-450 identified in the Robust03 track overview), and "combined" (all 250 topics). Note that new topics reflect blind evaluation while other topics do not. Besides usual metrics of mean-average precision (MAP) and precision-at-10 (P10), two non-standard metrics are reported which focus on difficult topics: "%no", referring to the percent of topics for which P10 = 0, and "area", referring to area under the MAP curve for the worst quarter topics. The latter two metrics were computed via a publicly available NIST script used in the original tracks: `http://trec.nist.gov/data/robust/robust2004_eval.pl`.

Chapter 3

Supervised Model Estimation with Regression Rank

This chapter presents a new supervised learning framework called "Regression Rank" for predicting effective term weights on novel queries given examples of past queries and their relevant documents. Term weights are generated by a feature-based model leveraging various statistics, and feature weights are learned from past queries via regression. We evaluate our approach with retrieval experiments on TREC `description` queries, which typically perform less accurately than `title` queries due to poor estimation of term weights. Experiments on three TREC collections show both improved search accuracy and significant potential for additional improvement.

3.1 Introduction

Classic approaches to IR have achieved broad success by exploiting highly-discriminating terms to model the relationship between queries and documents. Be it vector similarity (§2.4.1), document-likelihood (§2.4.2), or query-likelihood (§2.4.3), each adopts a simple bags-of-words representation, employs similar TF-IDF statistics [Zhai and Lafferty, 2004], and performs comparably in practice [Fang et al., 2004]. Despite their success, however, these classic approaches are limited by their common lack of support for supervised estimation: there is no mechanism by which term weight estimation can be improved over time. Often there is also no provision for inferring the relative importance of query terms in context of the specific query. We present an approach for tackling these related concerns in tandem.

Imagine we know for some past query that a particular word was important (i.e. assigning it high weight relative to other terms yields effective retrieval for that query). How does this knowledge inform our ability to weight terms effectively in future queries? The problem is effective term weighting is very much a context-sensitive issue, making it difficult to generalize anything about appropriate weighting from one query to the next. As a consequence, recent work in *learning to*

rank (LTR) [Joachims et al., 2007] has backed off from modeling individual words to instead employ aggregate measures of lexical compatibility. While using less specific features does enable cross-query learning, it comes at the cost of sacrificing model power to discriminate between relevant and non-relevant documents on the basis of individual terms. LTR methods offset this loss in lexical expressiveness by leveraging additional knowledge sources, but they leave us with the trade-off that learning can only be achieved by abandoning a hallmark strength of classic IR techniques.

In addition to having consequences for learning, context-sensitivity also limits retrieval effectiveness. For example, far improved retrieval accuracy has been achieved using relevance models to perform query-specific estimation of term weights [Lavrenko and Croft, 2001, Zhai and Lafferty, 2001]. The problem with addressing context-sensitivity via feedback-based approaches is that (explicit) relevance feedback requires user interaction, and pseudo-relevance feedback can be unreliable, depending on retrieval accuracy with the original query. As query length increases, the consequence of ignoring context also grow more severe. For example, consider TREC topic 331 shown in Table 2.2. Although the `description` contains additional informative terms in comparison to the `title`, these terms likely represent weaker correlations individually with the user's core information need. This means that despite being more informative, system failure to effectively weigh the importance of query terms yields a weaker understanding of the overall query, lowering retrieval accuracy.

Regression Rank represents a middle way between LTR and classic approaches, intended to capture the best of each: we can continue to leverage individual terms, learn term weights effectively from past queries, and incrementally add arbitrary features to smoothly transition toward richer query and document representations. Given a bag-of-words retrieval model (§3.2.1) and a set of training queries with relevant documents, we first estimate effective term weights for each query (§3.2.2). Because term weights do not generalize across queries, secondary features correlated with term weights are introduced to bridge this gap (§3.2.3). Finally, a regression function is learned on the basis of these features to predict term weighting for novel queries (§3.2.4). Though our presentation here restricts attention to lexical features (i.e. bag-of-words representation), our framework is as extensible as other LTR approaches in allowing arbitrary additional features to be incorporated into the retrieval model so long as one can correspondingly define secondary features for predicting retrieval model feature weights.

To evaluate our approach, we conduct retrieval experiments with TREC `description` queries on three TREC document collections (§3.3). While `description` queries present more challenging estimation requirements than `title` queries with respect to inferring query term weights, they also provide us with an opportunity to realize more accurate retrieval. Results show context provided by `description` queries enables us to realize more accurate search today and opens the door to largely untapped potential for additional improvement.

3.2 Method

This section describes Regression Rank's four components:

1. A retrieval model (parameterized uniquely for each query)
2. A procedure for estimating retrieval model parameters on a given query
3. A set of secondary features correlated with retrieval model parameters
4. A regression procedure to infer retrieval model parameters from features

3.2.1 The Retrieval Model

While the choice of retrieval model used with Regression Rank is largely unfettered, it must be drawn from some parametric family for which one can imagine corresponding secondary features correlated with the retrieval model's feature weights and generalizing across training examples (i.e. queries). We restrict work here to bag-of-words retrieval; our goal is to preserve the lexical discriminating power of classic approaches while augmenting them with two new strengths: the ability to effectively estimate query term weights given knowledge of past queries, and the ability to incrementally transition toward richer representations of queries and documents.

Of the three classic approaches, we adopt query likelihood for our retrieval framework. In this language model (LM) approach, we assume each observed document D is generated by an underlying LM parameterized by Θ^D. In §2.4.3, we showed how query-likelihood can also be interpreted as KL-divergence based ranking assuming the latent query unigram Θ^Q is estimated by maximum-likelihood (ML). Not only does this insight communicate the importance of effective query model estimation for achieving accurate search, but it also highlights query likelihood's implicit ML assumption that all query tokens are equally important.

In practice, some query terms will almost certainly correlate with the desired relevance distinction more than others, and this is particularly true with verbose, *natural* language queries in which many terms tend to individually represent weaker correlations. Unfortunately, the presence of these weaker correlations can have the undesirable effect of causing system focus to drift away from the user's core information need. Consequently, accurate estimation of Θ^Q necessarily plays a more significant role with verbose than keyword queries. However, complementing this challenge is a new opportunity. Assuming mass assignment to Θ^Q is restricted to terms observed in Q, verbose queries enable greater modeling power by projecting the discrimination task into a higher dimensional space in which the presence of additional terms provides greater flexibility for discriminating between relevance and non-relevance. The key here to both challenge and opportunity is effectively estimating Θ^Q.

We estimate Θ^D via standard Dirichlet-smoothing (§2.4.3) with a fixed hyper-parameter μ. Consequently, our estimation task is reduced entirely to effectively predicting the latent query unigram Θ^Q for novel input queries.

3.2.2 Estimating the Query Model

A key idea of Regression Rank is that one can generalize from effective query models (§3.2.1) of past queries to infer strong query models on future queries. To perform this generalization, we must have

query models to generalize from. This means we require a method for estimating the query model for each training query given examples of its relevant (and possibly non-relevant) documents.

There is a large body of related previous work to build on in relevance and pseudo-relevance feedback [Lavrenko and Croft, 2001, Zhai and Lafferty, 2001] as well as LTR [Joachims et al., 2007], but there are also some significant differences between our task here and that considered in previous work. Whereas relevance feedback usually assumes a handful of annotated examples, we can conceivably use every document in the collection for training. Pseudo-relevance feedback is noisy and depends on the accuracy of the initial retrieval. LTR training methods have directed little attention toward estimating lexical feature weights on a single query for a very good reason: there would be little point in the exercise. Since lexical features don't generalize across queries, we would simply be memorizing an optimal ranking for one particular query. However, this exercise becomes important in our framework.

We have adopted the classic strategy of grid search (cf. [Salton and Buckley, 1987]): sampling retrieval accuracy from a target metric space at regular points corresponding to candidate parameterizations. While this method of tuning has a long history in IR, some recent work in *learning to rank* has instead explored optimizing other surrogate functions in order to achieve more efficient training (cf. Joachims et al. [2007]). The tradeoff in doing so, however, is that a given surrogate function may be poorly correlated with the target metric one is actually interested in, leading to metric divergence [Metzler and Croft, 2007b]. While grid search scales relatively poorly as the parameter space increases, it is simple, reasonably efficient with few parameters and/or coarse sampling, and allows the retrieval metric of interest to be directly optimized. Our particular use of grid search has involved a couple of noteworthy details. First, grid search requires specifying the granularity of assignments to sample. We determined this via a reinterpretation of earlier work in query reduction [Kumaran and Allan, 2007]. This prior work generated all possible reductions (i.e. term subsets) of an original description query and then explored alternative methods of reduction selection. In the spirit of the earlier derivation (§3.2.1) in which query formulation was transformed into query model estimation, we let query reductions define the set of query models at which to evaluate retrieval accuracy in the metric space. Considering all reductions provides fairly robust coverage of the query model's effective assignment subspace. Because previous work showed most optimal query reductions contained six or fewer terms [Kumaran and Allan, 2008], we adopted an efficiency expedient and limited our sampling to query models containing six or fewer non-zero parameters.

The second noteworthy detail concerns how the query model is estimated once samples have been obtained from the metric space. The easiest solution is to simply pick the query model whose sample achieved maximum score on the target metric, but this may not be the best strategy in the context of Regression Rank. Recall our objective in inferring the query model is to enable eventual regression across queries (§3.2.4). The problem with this easy solution is that subsequent regression will be based on a single sample that may be drawn from a sharply-peaked local maximum on the metric surface. This would mean that were we to attempt to recover this parameterization via regression, small regression errors could yield a significant drop in metric performance. For this

reason, we instead estimate Θ^Q as the *expected* query model $\widehat{\Theta^Q} = \sum_s [\text{ Metric}(\Theta_s)\Theta_s]$, a sum in which each sample query model Θ_s is weighted by the retrieval accuracy it achieved. The intuition here is that this expectation should yield parameter values tending to perform well in general, and so the parameterization will more likely correspond to a smoother portion of the metric surface. Additionally, this strategy lends itself more easily to independent estimation of parameters since the expectation considers assignments to each parameter in a variety of contexts. A final detail is that since retrieval accuracy differed significantly across queries, the expected query model for each query was normalized to the interval $[0, 1]$ to provide a more stable basis for regression. Though we omit details, this yielded a statistically significant improvement on the development set (§3.3).

3.2.3 Secondary Features

Given examples of past queries and corresponding inferred query models, our next task is to identify secondary features. These features should both correlate with the query model and generalize across queries so that we may predict appropriate query models on future queries. This section describes our current feature set; a complete listing appears in Table 3.1. While existing features have proved effective, their paucity and simplicity can be taken as evidence that exploration of the feature space is far from complete.

Query model parameters can be understood as expressing relative term importance within the context of the overall query. As such, it should not be too surprising that the classic statistics of term frequency (*tf*) and document frequency (*df*) appear in our feature set (Features 1-12) to model term ubiquity and specificity, respectively. Since we are interested in relative rather than absolute term importance, we also computed these statistics in context of the other query terms (i.e. normalized) as well as in raw form. In addition to these classic statistics, we follow previous work (§2.7.2) to employ Google 1-gram *tf* [Brants and Franz, 2006] and *residual* inverse-*df* (*idf*) statistics (Features 13-14). The massive volume of the former is intended to provide another useful estimator of term frequency, particularly in the case of small collections, and the latter assumes important terms can be detected by distributional deviation from Poisson. While Google-based statistics provide a useful measure of term frequency on the Web, we also found it useful to gather the above collection-based statistics (i.e. *tf*, *idf*, and residual *idf*) from Gigaword [Graff et al., 2005] in addition to the target retrieval collection. This is reflected in Table 3.1's notating these feature *templates* as parameterized by a collection argument C to produce different feature *instances* for each collection. Use of out-of-domain data was motivated by previous work's empirical evidence of increased correlation between term importance and *idf* as collection size grows (§2.7.2), as well as another line of prior work having demonstrated significant retrieval benefit from leveraging external corpora (§7.2) [Diaz and Metzler, 2006]. A final traditionally-inspired feature, stop(q_i) (Feature 15), asks whether or not a given query term appears in the stop list (§3.3). While we do employ deterministic stopping, we stop before stemming to avoid accidental stemming collisions with the stop list. Nevertheless, stop words produced by stemming often are in fact unimportant to the query, and including a feature comparing stemmed words to the stop list proved useful.

Parameter	**Description**
$Q = q_1 \dots q_m$, i	Query Q of length m, indexed by i
C, N	Collection C containing N documents
n, w	Integer scalar & lexical token (parameters)
T	Part-of-speech tag-set

Feature Template	**ID**	**Type**	**Definition**
term frequency: $tf(C, Q, i)$	1	integer	tf_i: raw frequency of q_i in C
	2	real	$tf_i / \max_j^m tf_j$
	3	real	$tf_i / \sum_j^m tf_j$
	4	real	$log(tf_i)$
	5	real	$log(tf_i / \max_j^m tf_j)$
	6	real	$log(tf_i / \sum_j^m tf_j)$
document frequency: $df(C, Q, i)$	7	integer	df_i: # documents in C containing q_i
	8	real	$df_i / \max_j^m df_j$
	9	real	$df_i / \sum_j^m df_j$
	10	real	$log(df_i)$
	11	real	$log(df_i / \max_j^m df_j)$
	12	real	$log(df_i / \sum_j^m df_j)$
residual IDF: $ridf(C, q_i)$ (§2.7.2)	13	real	$log(N/df_i) - log(1/1 - e^{\alpha_i})$, $\alpha_i = tf_i/N$
Google TF: $gtf(q_i)$ (§2.7.2)	14	integer	raw frequency of q_i in Google 1-grams
stopword: stop(q_i)	15	boolean	is q_i a stopword?
q_i's location in Q: loc(i, m, n)	16	boolean	does $i = n$? (query initial)
	17	boolean	does $m - i = n$? (query final)
lexical info: context(Q, i, w)	18	boolean	does $q_{i-1} = w$?
	19	boolean	does $q_{i+1} = w$?
	20	boolean	is q_i trailed by comma?
part-of-speech: pos(q_i, T)	21	boolean	is tag(q_i) $\in T$

Table 3.1: Secondary features used to predict the query model. We define $log(0) \equiv 0$ and $\frac{anything}{0} \equiv 0$ to account for out-of-vocabulary query terms. Features are parameterized templates, instantiated with various settings to yield multiple feature instances.

Features 16-17 (location) correlate term importance with proximity to the start or end of the query string (experiments in §3.3 set $n = 5$ as the window size), and we found it beneficial to instantiate this feature for both the user's original query and its normalized version used in retrieval (i.e. after stopword removal, converting hyphenated compounds into separate terms, etc.). Features 18-20 (context) correlate term importance with presence of certain surrounding terms or punctuation. All possible terms were considered during feature collection, but few actually survived to instantiation due to feature pruning (see below). Feature 21 asks whether a given term's part-of-speech is a member of a given tag-set, correlating tag-sets with term importance[1].

Because a given statistic will be more reliably estimated under more frequent observation, we employed feature pruning to discard any instantiated feature that was not observed at least a parameter η times in the training data; we set $\eta = 12$ based on development set tuning (§3.3). As mentioned earlier, this significantly reduced the number of lexical features and generally helped filter out chance correlations from sparse features. Non-sparse features like *tf* which occur for every term were unaffected by pruning. Following previous work [Joachims et al., 2007], feature values were normalized to the interval $[0, 1]$.

Finally, the astute reader may have noticed our use of *df* rather than the more usual *idf* features and wonder if this was motivated. Consequently, we close this section with a quick note regarding *df* vs. $idf = N/df$ features in a linear regression model with bias term b (§3.2.4). Since N is constant, use of IDF simply varies feature j's bias contribution b_j and the sign of the learned weight λ_j: $\lambda_j log(N/df) = \lambda_j log(N) - \lambda_j log(df) = b_j - \lambda_j log(df)$. If all queries were of fixed length, bias contributions from all query terms could be equivalently folded into the model bias term, but varying query length requires an additional length feature in conjunction with *df* to achieve strict equivalence with *idf*. In practice, we saw little difference either way and our choice of df was arbitrary.

3.2.4 Inferring the Query Model via Regression

Given examples of target term weights paired with corresponding secondary features, our next task is to predict the query model based on the features. Since the output of our learner will be continuous values, our task is one of regression, and we follow a standard approach to accomplish it. That said, we will briefly motivate the approach taken and its merits in the context of our task.

Let N denote the number of query terms across the entire training set, $Y = \{y_1 \dots y_N\}$ the target term weights, and $\mathbf{X} = \{X_1 \dots X_N\}$ the feature vectors. Next, let d denote the number of features (i.e. dimensionality of our feature space) and $X_i = \{x_i^0, x_i^1 \dots x_i^d\}$ denote the ith feature vector with $x_j^0 = 1$ by definition for all j. Also, let $W = \{w_0 w_1 \dots w_d\}$ denote the weight vector (w_0 is the bias term). To define a learning objective, let $L(y, \hat{y}) = (y - \hat{y})^2$ define the loss function to minimize given a given target value y and our prediction for it $f(X, W) = \hat{y}$ (standard assumption of squared loss here assesses positive and negative errors equivalently). Assuming $\mathbf{X}$ and Y are drawn

[1] While the only part-of-speech distinction currently employed is distinguishing nouns and verbs from other categories, we actually fully parse the original query strings with a treebank parser [McClosky et al., 2006] once sentence boundaries have been detected [Reynar and Ratnaparkhi, 1997]. While tags might be more easily obtained without parsing, our use of parsing is intended to support future work exploring syntactic features.

from the joint distribution $p(X, y)$, our goal is to minimize risk (i.e. expected loss, c.f. [Lafferty and Zhai, 2001]): $\mathbb{E}_{(X,y)\backsim p}[L(f(X, W), y)]$. Lacking oracle knowledge of $p(X, y)$, we approximate risk with the empirical loss $\sum_i^N L(f(X_i, W), Y_i) = \sum_i^N (y_i - \sum_{j=1}^d w_j x_i^d)^2 = (Y - \mathbf{X}W)^T(Y - \mathbf{X}W)$ and seek an optimal weight vector W^* minimizing this quantity. Conveniently, this *sum of least squares* optimization problem has a closed form solution: $W^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$.

However, W^* above corresponds to the setting of W maximizing the likelihood of Y assuming it is generated by adding Gaussian noise to $f(X, W)$. Recalling our earlier discussion regarding maximum likelihood (ML) estimation of the query model(§3.2.2), we saw that ML's accuracy suffers from its estimate being uninformed by prior knowledge. Here we see another aspect of this same problem: ML acts as a *what-you-see-is-all-there-is* estimator, assuming any observational knowledge provided is complete. In particular, it suffers from over-fitting, assigning probability mass to weight features which may be demonstrating only chance rather than true correlation. To rectify this, one often regularizes the optimization by penalizing larger weight vectors in the loss function. For example, we could revise our empirical loss formulation as $\sum_i^N L(f(X_i, W), Y_i) = (Y - \mathbf{X}W)^T(Y - \mathbf{X}W) + \beta W^T W$ where β defines a controllable regularization parameter. This regularization of the ML solution is known as ridge (or L2) regression and also has a convenient closed-form solution: $W^* = (\beta I + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$ where I denotes the identity matrix. Another alternative, lasso (L1) regression, penalizes the absolute value of W is instead of its square. While lasso regression does not have a closed-form solution, many techniques exist for computing it, and its stiffer penalization of W can lead to sparser assignments with fewer spurious features.

In empirical trials comparing the three techniques described above, (ML, L1, and L2 regression) with respect to squared loss on the development set (§3.3), L2 consistently performed best, with manual sweep of β finding an optimal setting at $\beta = 1$. Consequently, we adopt this approach in our retrieval experiments. We also evaluated logistic regression, which would be more principled to employ here since the output Θ^Q really ought to be constrained to being a probability distribution, but we saw little empirical difference in practice (recall the model is invariant to parameter scaling). Other regression variants were also evaluated but not found to be sufficiently remarkable in comparison to merit discussion.

3.3 Evaluation

This section evaluates effectiveness of Regression Rank on three TREC collections of varying size and content (Table 2.4). All model development was performed on the Robust04 collection using 149 topics (301-450 except 342 due to its excessive length); remaining topics (including 342) and collections were reserved for blind evaluation. Final results (Table 3.2) use all available data.

Keyword and description queries were taken from topic `title` and `desc` fields, respectively[2]. Model training used 5-fold cross-validation, and retrieval was performed using Indri [Strohman

[2]While TREC evaluations often use `title` and `desc` fields concatenated as an evaluation condition, the resultant queries do not read naturally and artificially reinforce key terms through repetition. In contrast, the `desc` field alone provides a more realistic example of an information need expressed in *natural* language.

et al., 2004]. Primary evaluation metrics were mean-average precision (MAP) and top-5 precision (P@5), as reported by `trec_eval` 8.1[3]. Results are marked as significant† ($p < 0.05$), highly significant‡($p < 0.01$), or neither according a non-parametric randomization test computed by Indri's `ireval` [Smucker et al., 2007]. Experimental conditions were designed to match previous work (§2.7.2) for comparison. Queries were stopped at query time using the same 418 word INQUERY stop list [Allan et al., 2000] and then similarly Porter stemmed [Porter]. The same Dirichlet parameter $\mu = 1500$ (§3.2.1) was used.

We begin by presenting development set (Table 2.4) results. As a baseline, we evaluate the standard practice of inferring the query model by maximum-likelihood (ML), i.e. assigning uniform weight to each token observed in the query. Under this baseline, keyword queries achieve 2.83%‡ higher MAP (absolute) than description queries. Since additional terms introduced by description queries tend to individually represent weaker correlations with the desired distinction between relevant and non-relevant documents, intuitively these terms should be assigned lower weight in the inferred query model. The baseline fails to do this, however, and retrieval accuracy falls as a result. In contrast, using Regression Rank with description queries improves accuracy 4.17%‡ over baseline description results and 1.34% over baseline keyword results.

Using TREC topics to compare accuracy achieved with keywords and description queries, analyzing the effect of query verbosity is occasionally complicated by important keyword terms missing from the descriptions. In these cases, keyword queries may benefit from being more informative in addition to being tightly focused. To control for this, we identified 122 development set topics for which all keywords were contained in the descriptions and evaluated performance on this topic subset. Baseline keyword accuracy improvement over description accuracy was reduced to 2.31%† (absolute). Further, Regression Rank achieved 4.54%‡ over baseline description and 2.23%‡ over baseline keyword results. Figure 3.1 shows change in retrieval accuracy over all development set topics as a function of query length. Broad improvement is seen both in terms of number of queries improved and change in MAP at each length.

Recall the first step in our framework is to estimate importance of terms in each training query given that query's relevant documents, and recall that we accomplish this by sampling retrieval accuracy achieved under different candidate query models (§3.2.2). Given that subsequent regression is based on the estimated query models, intuition suggests more accurate estimation should yield more accurate retrieval following regression. To test this, we tried restricting sampling to queries of 15 words or less, reducing the total number of samples from 502K to 104K. When performing regression based on this reduced sample set, retrieval accuracy fell 1.07%‡ (absolute). While these results are certainly sensitive to the sampling procedure used, it nonetheless seems clear that strong estimation of training query models has an important effect on downstream retrieval accuracy. This further suggests additional gains might be realized by implementing a more effective estimation procedure or simply increasing the number of samples taken.

Our main results (Table 3.2) use all queries for all three TREC collections (Table 2.4). In addition

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

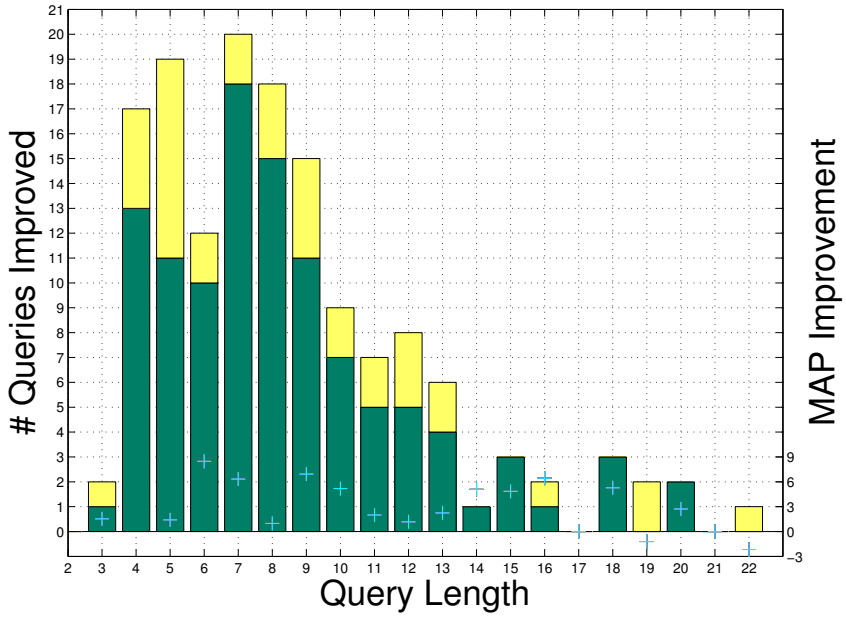

Figure 3.1: Analysis of development set results shows retrieval accuracy improvement as a function of query length. Bars show the number of queries for each query length and the ratio of them that were improved. MAP improvement achieved at each length is marked by '+'.

Results

		Robust04				W10g				GOV2			
Query	**Model**	P@5	Δ	MAP	Δ	P@5	Δ	MAP	Δ	P@5	Δ	MAP	Δ
Title	ML	47.55		24.97		31.80		19.61		55.97		29.81	
Desc.	ML	47.31	-	24.29	-	40.00	-	18.49	-	51.81	-	25.42	-
	RRank	$51.97^{\ddagger}_{\ddagger}$	4.66	$27.09^{\ddagger}_{\ddagger}$	2.80	$41.40^{\ddagger}$	1.40	$22.15^{\dagger}_{\ddagger}$	3.66	53.96	2.15	$27.42_{\ddagger}$	2.00
	*REG	60.16	12.85	32.01	7.72	46.60	6.60	27.95	9.46	62.60	9.79	33.43	8.01
	*RED			35.07	10.78			31.75	13.26			36.03	10.61

Key Concepts (§2.7.2) [Bendersky and Croft, 2008]

		Robust04				W10g				GOV2			
Query	**Model**	P@5	Δ	MAP	Δ	P@5	Δ	MAP	Δ	P@5	Δ	MAP	Δ
Title	ML	47.80		25.28		30.73		19.31		56.75		29.67	
Desc.	ML	47.26	-	24.50	-	39.20	-	18.62	-	52.62	-	25.27	-
	◇SDep	49.11	1.85	25.69	1.19	39.80	0.60	19.28	0.66	56.88	4.26	27.53	2.26
	KCon	48.54	1.28	26.20	1.70	40.40	1.20	20.46	1.84	56.77	4.15	27.27	2.00

Table 3.2: Retrieval results comparing methods for term weight estimation using all queries and collections (Table 2.4). A baseline maximum-likelihood (ML) technique is compared to Regression Rank (RRank) and the Key Concepts (KCon) model (§2.7.2). Primary comparisons are shaded. Results from a non-unigram dependency model (◇SDep) reported previously [Bendersky and Croft, 2008] are also shown. Since our baseline results differ slightly with those reported earlier [Bendersky and Croft, 2008], we present both sets of baselines to show each work's improvement Δ relative to its own baseline. Oracle runs show retrieval accuracy under conditions of perfect regression (*REG) and perfect reduction (*RED). Score^{t}_{d} superscript and subscript annotations indicate significance with regard to title and description baselines.

to the ML baseline defined earlier, we also compared to Bendersky and Croft's Key Concepts model (§2.7.2). Regression Rank achieved highly significant MAP improvement over baseline description accuracy for all collections. Compared to baseline keyword accuracy, MAP improvement was highly significant for Robust04 and significant for W10g; both Regression Rank and Key Concepts failed to improve over baseline keyword accuracy. Regression Rank also achieved 1.1% and 1.82% absolute MAP improvement over Key Concepts for Robust04 and W10g, with equal performance achieved on GOV2[4].

Another baseline strategy one might consider for verbose queries is to simply ignore all terms but the nouns. In other words, automatically part-of-speech tag input queries and employ a stoplist filtering out non-noun categories (rather than the more usual practice of stopping specific terms). We did not evaluate this baseline, but Bendersky has reported it achieves poor accuracy in practice due to the importance of non-nouns in some queries[5]. This approach would also suffer from the same problems with robustness as typical term-based stopping (§8.1).

We also report retrieval accuracy under two oracle conditions: perfect regression and perfect reduction. Perfect regression indicates retrieval accuracy that would be achieved if we could perfectly recover the expected query models estimated by sampling (§3.2.2). This provides an indication of how well secondary features and regression are working, how well our regression strategy can work across collections, and the potential for future improvement by improving features and regression. Perfect reduction results are higher and indicate accuracy achieved by the best query reduction found for each query during sampling. This result shows that even if perfect regression were achieved under current conditions, significant further improvement would still be possible if we could perform accurate regression on the basis of optimal reductions instead of expected query models. However, this would present a further challenge to regression since expected query models are more stable against regression error (§3.2.2).

3.4 Discussion

While Key Concepts (§2.7.2) presents a similar learning strategy, several important differences differentiate the two approaches. With Key Concepts, learning is applied only for predicting noun phrase (NP) weights. This means all terms within a noun phrase are assigned the same weight from the learning component (i.e. tied parameters), and mixture with the ML estimate is required to weight all other terms. In contrast, Regression Rank's learning predicts all term weights without any parameter tying.

Another notable difference is in terms of the form of supervision employed. Key Concepts learns from NP annotations whereas Regression Rank uses document relevance annotations. The advantage of the former approach is that the annotation task is likely simpler and therefore cheaper if one wanted to pay for additional annotations in order to further improve learner accuracy. However,

[4] Statistical significance was not measured here due to lack of access to Key Concepts rankings at the time.

[5] Personal communication

since document relevance annotations are widely produced to facilitate system evaluation already, we can exploit these annotations serendipitously for supervised learning on today's and tomorrow's document collections at no cost. Even better, document relevance-based learning is directly amenable to estimation via click-through data [Joachims, 2002] rather than being constrained by manual annotation. Instead, lifetime model learning is possible by harnessing the ever-growing logs of search engine use and letting click redundancy compensate for uncertainty of relevance.

An additional benefit of Regression Rank's learning strategy is that it directly maximizes the target metric of interest (§3.2.2). In contrast, Key Concepts learning is indirect, trusting classifier confidence in reproducing human intuition of key NPs will translate to improved retrieval accuracy.

Other important related work to mention includes term-specific smoothing [Mei et al., 2007] and probabilistic indexing [Fuhr and Buckley, 1991]. The former method also attempts to unlock the expressiveness of term-based models via better estimation, but it tackles a much more difficult estimation problem by trying to directly learn one parameter per term. We simplify this problem by estimating a much smaller secondary feature space, and this further lets us predict context-dependent parameters for terms. Probabilistic indexing also applied supervised learning but carried it out for documents rather than the query, used a much smaller feature space, and was evaluated on a relatively small datasets by today's standards.

Overall, we believe the Regression Rank learning framework exhibits several useful properties:

- **two-layer hierarchical modeling**: We model each problem instance via instance-specific "primary" features, then generate primary features via "secondary" features generalizing across problem instances. In short, we exploit a level of indirection in modeling the problem.
- **metric and model independence**: Since the framework can maximize an arbitrary target metric atop any parametric model, it can be broadly applied as a general learning procedure.
- **separation of concerns**: Since estimation is performed atop an arbitrary parametric model, we can preserve wholesale existing methodology and engineering effort for achieving efficient, scalable search. Research on both fronts can proceed in parallel with minimal interaction.
- **incremental extensibility**: Independence of model and estimation mean we can explore incremental additions to the feature space without needing to revise the learning procedure.
- **context-dependent modeling**: We can learn context-dependent weights over context-independent search model features by capturing context in secondary features rather than the search model. This yields a novel approach for achieving query-dependent feature or model combination [Geng et al., 2008].

3.5 Future Work

The estimation approach we adopted directly maximized target metric retrieval accuracy in order to avoid divergence between optimization and evaluation goals (§3.2.2). However, it should be noted

that an alternative form of metric divergence still exists in our method via our use of regression (§3.2.4). This is seen by observing that our regression procedure minimizes squared loss of data fit rather than maximizing our target IR metric. A possible strategy for ameliorating this effect would be to perform regression based on *all* parameterizations evaluated during grid search rather than just the expectation computed over those samples (§3.2.2). In other words, whereas current regression training for each term targets a single scalar y_i weight given the term's feature representation X_i, instead we would have a target vector of weights $\overrightarrow{y_i}$ and the corresponding metric accuracy each achieved $\overrightarrow{a_i}$. Such weighted regression is easily performed using Matlab's "lscov" method or a similar utility. The intuition here is that rather that perform inference off a single point estimate (i.e. the expectation), we would likely be better served to propagate uncertainty regarding optimal parameterization to the regression module so that it can be taken into account there. Of course this would still rest on an approximation since the metric accuracy for each instance is not purely a function of the individual term and its weight, but rather a joint score over for the query as a whole, and it may be that a particularly high scoring joint parameterization is sensitive to settings of all parameters. However, performing weighted regression across all (weight,accuracy) pairs for a given term should amortize such effects as accomplished explicitly by our original use of the expectation.

3.6 Conclusion

This chapter introduced a supervised learning framework called "Regression Rank" for predicting effective term weights on novel queries given examples of past queries and their relevant documents. Term weights were generated by a feature-based model leveraging various statistics, and feature weights are learned from past queries via regression. We evaluated our approach with retrieval experiments on TREC `description` queries, which typically perform less accurately than `title` queries due to poor estimation of term weights. Experiments on three TREC collections showed both improved search accuracy and significant potential for additional improvement.

In Ch. 4, we describe how Regression Rank can be applied to achieve more effective Markov Random Field modeling (§2.6), and Ch. 5 shows how the feature spaced employed here (§3.2.3) can be greatly simplified while still maintaining equivalent retrieval accuracy. Subsequent chapters do not apply Regression Rank but explore related issues: coping with verbosity of documents for effective relevance and pseudo-relevance feedback (Ch. 6), and improving estimation with sparse document collections (Ch. 7).

Chapter 4

Better Markov Random Field modeling

The previous chapter introduced Regression Rank, a supervised learning framework for retrieval model estimation, and showed how it could be used to improve term-based retrieval accuracy for verbose queries. In this chapter, we study the combined effect of applying our more effective term-based model in conjunction with modeling term adjacency and proximity features. In particular, we use Regression Rank to better estimate the unigram component of Metzler and Croft's Markov random field (MRF) model [Metzler and Croft, 2005] (§2.6). While the original MRF formulation includes a parameter for each of its three feature classes (i.e. terms, adjacency, and proximity), parameters within each class are set via a uniform weighting scheme adopted from the standard unigram. We hypothesize greater MRF retrieval accuracy can be achieved by better estimating these within-class parameters, and we empirically test this hypothesis using the same document collections and verbose queries reported on in the previous chapter. Results show improved estimation of MRF's unigram component consistently out-performs both the MRF's baseline performance and our supervised unigram results from the previous chapter. We further study the interaction of these approaches with pseudo-relevance feedback [Lavrenko and Croft, 2001]. Finally, we present additional results demonstrating the potential benefit to be realized by better estimating MRF term interaction parameters.

4.1 Introduction

Classic term-based approaches rank documents using a linear model computed over a feature space of lexical terms (often coupled with a document-specific prior) [Ponte and Croft, 1998, Singhal et al., 1996, Sparck Jones et al., 2000]. This simple feature set is remarkably expressive: a vast number of rankings are possible given different settings of the individual term weights. In contrast to this modeling expressiveness however, successful strategies for estimating term weights have relatively

limited (though many variations have been explored over the years, often in an ad hoc fashion). Lack of statistical learning means estimation accuracy cannot automatically improve as more observational evidence becomes available. Recent work in supervised estimation of these models, such as described in the previous chapter, has sought to remedy this deficit.

Of course, language conveys far more information than simple term-based models are able to capture, and an important goal for long-term research is to develop richer models of language. A recent contribution in this direction was the development of a Markov random field (MRF) approach in which a standard unigram model is supplemented by two additional classes of lexical features: contiguous phrases and proximity (§2.6). While this approach was certainly not the first to use phrases or proximity (cf. [Brin and Page, 1998, Clarke et al., 2000, Gao et al., 2004, Mishne and de Rijke, 2005] inter alia), it incorporates them via a simple, principled framework that is efficient to compute and has been shown to consistently out-perform the standard unigram model across a range of TREC document collections [Bendersky and Croft, 2008, Metzler and Croft, 2005]. An important detail of the approach, however, is that although the weights for each feature class are learned from data, feature weights within each class are in fact estimated by the same uniform assumption as the standard unigram. This means that MRF estimation is similarly limited in modeling the varying importance of query terms. Recognizing this limitation, however, also reveals a potential opportunity to improve MRF accuracy by employing a similar supervised approach for parameter estimation as has already been successfully applied to unigram modeling (Ch. 3).

In this chapter, we show this strategy is indeed viable: retrieval accuracy of the MRF model can be significantly increased by applying supervised learning. We evaluate retrieval for verbose queries in particular in order to improve document retrieval underlying question answering and other focused retrieval tasks [Kamps et al., 2008]. Our main results show that in comparison to using either the original MRF approach (§2.6) or a supervised unigram model (Ch. 3), integrating supervised unigram model estimation into the MRF yields significantly improved retrieval accuracy for verbose queries across three TREC document collections (§4.3.2). Additional experiments performed show the strength of our improved MRF under blind-feedback as well (§4.3.3). Finally, we evaluate model performance under optimal weighting of phrase and proximity features to demonstrate how their more accurate estimation also significantly improves retrieval (§4.3.4). This last experiment shows 3% absolute improvement over the baseline model can be achieved by assigning all phrasal and proximity weight to a single key dependency. In total, our results provide strong evidence that more accurate estimation of feature weights within each lexical class can significantly impact MRF model effectiveness. Results also motivate additional work exploring supervised estimation of feature weights for phrasal and proximity features alongside those of individual terms.

4.2 Method

§2.6 summarized Metzler and Croft's Markov Random Field model for retrieval (§2.6) and showed that its term-based feature class computes the standard Dirichlet-unigram. This means that the

feature class embodies the same implicit ML assumption that was shown earlier to underly the unigram model (§2.4.3). Moreover, since all three of the MRF's feature classes can be expressed in the same generic form, phrasal and proximity classes also make the same ML assumption. In other words, all observed feature instances for each class are assumed to be equally important in estimating the form of the latent information need corresponding to each class. Another way to say this is that all features within the same feature class are weighted by the same tied parameter λ_i. This reflects a choice of potential functions used rather than a general limitation of MRF modeling. We can generalize the model by instead assuming a unique potential function $\psi_i^c(c)$ for each clique rather than having a single function $\psi_i(c)$ for each feature class: $\psi_i(c) = \lambda_i \sum_{c \in i} \psi_i^c(c) = \sum_{c \in i} \lambda_i^c f_i^c$. The class-wide weighting parameter λ_i is preserved here simply for convenience. This generalized model is equivalent to the original under the condition that all clique-specific potential functions $\psi_i^c(c)$ within the same feature class adopt the same statistic S_i and use the same tied parameter $\lambda_i^c = \frac{1}{|c \in i|}$. We argue for breaking this parameter tying and applying supervised learning to estimate a unique weight for each clique to better model context-sensitivity.

Estimation with Regression Rank

We have just discussed how the MRF term component computes the standard Dirichlet-smoothed unigram. Consequently, Θ^Q is implicitly estimated by ML in the MRF as well to yield a uniform distribution over Q's terms. For example, we saw above that each clique is implicitly assigned uniform weight $\frac{1}{|c \in i|}$. This is problematic for verbose queries in which many terms appearing in the query are not strongly related to the core information need and should be assigned lower weight to improve retrieval effectiveness [Bendersky and Croft, 2008, Kumaran and Allan, 2007]. A similarly striking effect for dependencies is observed in §4.3.4.

Fortunately, we saw in Ch. 3 that Θ^Q could be more accurately estimated by applying supervised learning. Instead of applying the MRF's default ML estimation of Θ^Q, we instead use Regression Rank. We adopt the generalized MRF having a distinct $\psi_T^c(c)$ for each clique; the same term frequency statistic is used across terms but the parameter λ_i^c is not tied. We then use our supervised estimate of Θ^Q to set λ_i^c values. This yields a more effective term component in the MRF with the potential of improving the overall MRF ensemble's retrieval accuracy. We evaluate this in §3.3.

While we do not apply supervised estimation of phrasal f_O and proximal f_U feature weights here, results in §4.3.4 motivate future work in this direction. This might be achieved, for example, by applying Regression Rank to predict MRF rather than unigram parameters and extending its secondary feature set accordingly. In §4.4, we further discuss how the MRF model can be generalized beyond ways in which it has been historically used, as well as how better estimation of its parameters can enable us to take greater advantage of its full modeling power.

4.3 Evaluation

This section presents empirical results measuring the impact of better MRF model estimation on document retrieval accuracy. Retrieval experiments are conducted using the same three TREC document collections as in Ch. 3 (Table 2.4). In order to improve document retrieval for verbose queries like those found in question answering and other focused retrieval tasks [Kamps et al., 2008], evaluation primarily addresses use of TREC description queries (also as in Ch. 3). We use the *sequential dependence* MRF in our work since the *full dependence* MRF's combinatorial feature growth renders it intractable for use with verbose queries. An interesting topic for future work will be performing feature selection over all dependencies, sequential and non-sequential alike (§4.4).

As in Ch. 3, documents are ranked using Indri [Strohman et al., 2004], with rankings scored using `trec_eval` 8.1[1]. Mean-average precision (MAP) serves as the primary metric, and results are marked as significant† ($p < 0.05$), highly significant‡($p < 0.01$), or neither according a non-parametric randomization test computed by Indri's `ireval` [Smucker et al., 2007]. Experimental conditions reproduce those of previous work [Bendersky and Croft, 2008] and Ch. 3 for fair comparison. Queries were stopped at query time using the same 418 word INQUERY stop list [Allan et al., 2000] and then Porter stemmed [Porter]. The same Dirichlet hyper-parameter $\mu_T = 1500$ was used for term features as well as Indri default values for μ_O and μ_U phrasal and proximity hyper-parameters. A window size of 8 tokens was used with the proximity feature.

4.3.1 Estimating MRF Component Weights

Recall that the MRF model uses three classes of lexical potential functions: individual terms $\psi_T(c)$, contiguous phrases $\psi_O(c)$, and proximity $\psi_U(c)$ (§2.6). These potential functions are parameterized by λ_T, λ_O, and λ_U weights specifying the relative importance of each lexical class in the overall MRF ensemble. In the original work (§2.6), grid search was used estimate class weights using title queries over several document collections. Results showed an 85-10-5 mixing ratio (i.e. $\lambda_T = 0.85$, $\lambda_O = 0.10$, and $\lambda_U = 0.05$) generally performed well across collections.

We begin our evaluation by testing the optimality of these recommended λ_T, λ_O, and λ_U settings for verbose queries since earlier work applied the MRF's 85-10-5 mixing ratio to them without testing it (§2.7.2, Ch. 3). In comparison to title queries, verbose queries also exhibit more frequent syntactic relations between adjacent terms, and semantically-related terms often occur farther apart. Furthermore, the greater effectiveness of the supervised unigram in comparison to the maximum-likelihood (ML) unigram model used in the original MRF experiments suggested the unigram component here might merit additional weight in the mixture.

Consequently, we performed our own grid search over possible mixture ratios using development topics (§4.3.3). Despite any premonitions to the contrary, the 85-10-5 mixing ratio achieves MAP performance remarkably close to optimal: 24.79 vs. 24.93 for Robust04, 23.18 vs. 23.35 for W10g, and 26.68 vs. 27.01 for GOV2 (significance not reported). We therefore adopt the 85-10-5 ratio in

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

Query	Model	Robust04	W10g	GOV2
Title	Base unigram	25.32	19.49	29.61
Desc.	Base unigram	24.51	18.61	25.22
	MRF (§2.6)	25.64	19.14	27.40
	Key Concepts (§2.7.2)	25.91	20.40	27.44
	Regression Rank (Ch. 3)	27.33	22.01	27.35
	MRF + Regression Rank	$28.48^{\ddagger}_{\ddagger}$	$23.05^{\ddagger}_{\dagger}$	$29.51^{\ddagger}_{\ddagger}$

Table 4.1: Main results compare MAP retrieval accuracy of baseline MRF [Bendersky and Croft, 2008] and Regression Rank (Ch. 3) models vs. their combination. Score^{m}_{r} superscripts and subscripts indicate statistical significance of the combined model vs. the MRF (m) and Regression Rank (r) baselines. Key Concepts (§2.7.2) and canonical unigram accuracy are also reported.

Query	Model	Robust04	W10g	GOV2
Title	Base Unigram	48.11	31.20	56.24
Desc.	Base Unigram	47.63	39.20	52.21
	MRF (§2.6)	49.32	38.80	56.38
	Key Concepts (§2.7.2)	47.55	41.40	57.05
	Regression Rank (Ch. 3)	52.05	40.60	54.50
	MRF + Regression Rank	$54.30^{\ddagger}_{\ddagger}$	$42.00^{\dagger}_{\dagger}$	57.18

Table 4.2: Precision at top 5 ranks corresponding to same retrieval experiments as in Table 4.1.

our subsequent experiments for convenient comparison to previous work.

4.3.2 Estimating Term Feature Weights

This section presents our main results (Table 4.1) evaluating retrieval accuracy of the original MRF (§2.6), Regression Rank unigram (Ch. 3), and our combined model. Following previous work, Regression Rank was trained on each collection using 5-fold cross-validation. However, since the model was developed using only Robust04 (topics 301-450), further improvement of its performance and that of our combined model may also be possible for W10g and GOV2 collections via collection-specific model tuning.

Baseline performance of a standard unigram estimated by ML for both title and description queries shows that title queries consistently perform better than their description counterparts under ML estimation. While description queries are more informative to a human reader, additional terms introduced relative to title queries tend to individually correlate more weakly with the query's underlying core information need. Consequently, these terms should generally be assigned lower weight in estimation. ML's assumption that all observed query terms are equally important fails to do this, and retrieval accuracy suffers. The supervised estimation of Key Concepts (§2.7.2) and Regression Rank (Ch. 3) models addresses this limitation and is able to improve unigram retrieval accuracy as a result.

Our combined MRF model further exploits this better unigram estimation by leveraging it in

Model	Robust04		W10g		GOV2	
	Test	All	Test	All	Test	All
MRF (§2.6)	38.92	30.09	19.99	20.02	32.37	30.26
Regression Rank (Ch. 3)	37.03	30.52	21.77	22.48	30.36	28.96
MRF + Regression Rank	$39.13_{\ddagger}$	$31.82^{\ddagger}_{\ddagger}$	$23.19^{\dagger}_{\dagger}$	$23.05^{\ddagger}$	$32.91_{\ddagger}$	$31.20_{\ddagger}$

Table 4.3: MAP accuracy achieved by MRF (§2.6), Regression Rank (Ch. 3), and combined models for test and all topics using pseudo-relevance feedback. Statistical significance is reported as in Table 4.1.

conjunction with phrasal and proximity features. Across the three collections (Robust04, W10g, and GOV2), the combined model achieves absolute MAP improvements of 2.84% ‡ ($p < 0.0001$), 3.91% ‡ ($p = 0.0003$), and 2.11% ‡ ($p = 0.0003$) respectively vs. the original MRF. The number of queries improved, hurt or unchanged for each collection respectively are 166/83/0, 67/31/2, and 96/52/1. In comparison to the Regression Rank supervised unigram (Ch. 3), absolute MAP improvements of 1.15% ‡ ($p < 0.0001$), 1.04% † ($p = 0.0282$), and 2.16% ‡ ($p < 0.0001$) are achieved. In this case, number of queries improved, hurt or unchanged are 151/96/2, 50/48/2, and 82/66/1.

Precision at early ranks also shows signs of improvement. For the top-5 retrieved documents, the combined model achieves absolute improvements of 4.98% ‡ ($p = 0.0001$), 3.20% † ($p = 0.0329$), and 0.80% respectively vs. the original MRF for Robust04, W10g, and GOV2, respectively. The number of queries improved, hurt or unchanged for each collection are 73/37/139, 32/17/51, and 36/38/85. In comparison to the Regression Rank supervised unigram (Ch. 3), absolute precision improvements of 2.25% ‡ ($p = 0.0042$), 1.40%, and 2.68% are achieved. Here, the number of queries improved, hurt or unchanged are 52/29/168, 22/16/62, and 35/24/90.

4.3.3 Pseudo-relevance Feedback

This section reports retrieval accuracy of the original MRF model (§2.6), Regression Rank (Ch. 3), and our combined model under pseudo-relevance feedback (PRF) (§2.5.2, §2.6.2). PRF was performed using Indri [Strohman et al., 2004], which implements a variation on Lavrenko's relevance models [Lavrenko and Croft, 2001]. Only unigram feature weights are re-estimated via PRF since previous work saw little benefit from PRF for re-estimating dependency feature weights [Metzler and Croft, 2007a]. Ten feedback documents were used, with estimated feedback document models truncated to the most probable 50 terms per document. The feedback model mixture weight was tuned on development topics: 301-450 for Robust04, 451-500 for W10g, and 701-750 for GOV2. This resulted in feedback model weights of 0.6, 0.1, and 0.3 for the three collections. Primary evaluation was performed on the remaining topics. Results appear in Table 4.3. Accuracy on all topics is also shown and allows comparison to earlier non-PRF results (Table 4.1). Earlier Tables 2.5 and 2.6 show accuracy of the MRF run on Robust04 topic subsets.

For test set topics across the three collections, MAP accuracy of the combined model was improved by 2.10% ‡ ($p = 0.0001$), 1.42% † ($p = 0.0338$), and 2.55% ‡ ($p = 0.0001$) absolute vs.

Regression Rank. The number of queries improved, hurt, or unchanged for each collection were 64/33/2, 24/26/0, and 58/41/1. In comparison to the baseline MRF model, MAP increased by 0.21%, 3.20% † ($p = 0.0252$), and 0.54%, with the number of queries improved, hurt, or unchanged being 44/55/1, 30/20/0, and 49/50/1. As for comparative precision at early ranks, we briefly summarize results. For the top-5 retrieved documents, differences are not significant with respect to the base MRF, but the combined model does achieve significantly better precision than Regression Rank across all collections (highly significant for Robust04).

Over all topics, the combined model is also seen to consistently perform best. While highly significant MAP improvement is achieved over both MRF ($\Delta = 1.73\%, p = 0.0012$) and Regression Rank ($\Delta = 1.30, p < .0000$) for Robust04, we see an alternation of highly significant improvement over MRF for W10g ($\Delta = 3.03, p = 0.0013$) and over Regression Rank for GOV2 ($\Delta = 2.24, p = 0.0001$) due to Regression Rank performing better for W10g while the base MRF model performs better for GOV2. Lacking a means of predicting which base model will perform better for which collection under PRF, the combined model is attractive in providing insulation from this alternation, performing at least as well as the stronger base model in either case. When both base models do perform well (e.g. Robust04), the combined model is seen to out-perform both of them.

4.3.4 Phrasal and Proximity Feature Weights

Thus far, results have addressed the impact of better estimating MRF term weights. We now report the impact of better estimating MRF phrasal and proximity parameters.

Previous work generating all possible term subsets of verbose queries found retrieval accuracy could often be far improved by reducing queries to six or fewer terms (§2.7.2). This inspired us to try a similar experiment for phrasal and proximity features (i.e. sequential dependencies). We evaluated dependency reductions of the base MRF model in which the default set of all sequential dependencies was similarly reduced to a subset of at most six dependencies. This is equivalent to performing a grid search exploring possible binary assignments to these parameters (cf. [Salton and Buckley, 1987]). Other standard settings of the base MRF were kept fixed: 85-15-5 component weights along with the ML unigram weighting scheme.

Results in Table 4.4 show retrieval accuracy on Robust04 using a set of development topics (301-450). Statistical significance is not reported but can be safely assumed for the magnitude of improvements we discuss. The most striking observation is that inclusion of only the single most-helpful dependency improves MAP accuracy almost 3% absolute vs. the baseline model's default inclusion of all dependencies (i.e. ML estimation of dependency parameters). Furthermore, we see that adding a second best dependency provides no additional benefit, and that use of any greater fixed-sized subset of dependencies only serves to hurt performance vs. use of the single best dependency. Previous work modeling individual terms has similarly found that emphasizing one or two key terms in verbose queries also has the most significant impact on unigram retrieval accuracy [Bendersky and Croft, 2008]. It would be interesting to measure the degree to which key terms predicted in that work overlap with key dependencies found here. Results also show that if

Dependencies	MAP	P@5
all (baseline)	21.10	43.84
1-best	24.02	50.27
2-best	24.05	51.37
3-best	23.67	51.10
4-best	23.11	49.18
5-best	22.73	48.49
6-best	22.27	47.12
oracle	25.49	55.07

Table 4.4: MAP retrieval accuracy of MRF model (§2.6) under varying parameterization of phrasal and proximity features. The Robust04 collection was used with 146 description queries of length 20 or less (topics 301-450). Parameterizations were restricted to binary assignments of pair-wise sequential dependencies. Statistical significance is not shown.

it were possible to simply identify the group of six most helpful dependencies without regard to their respective ordering, improvement of 1% could still be achieved vs. the baseline. Finally, we see upper-bound improvement of about 4% could be achieved by picking the optimal number of best dependencies to use for each query.

Several details merit further optimism regarding the retrieval benefit of better estimating phrasal and proximity parameters. The grid search we performed considered only sequential dependencies; feature selection or weighting over the full cross-product of query dependencies (i.e. the full-dependency model) can only improve upon these results. Similarly, our grid search was restricted to binary assignments of parameters; more flexible weighting might also yield greater improvement. We also assumed fixed MRF component weights and ML estimation of phrasal and proximity parameters; additional relaxation of these assumptions may increase accuracy further. Previous work on sentence retrieval has also shown statistics regarding co-occurrence and syntactic relationships can be usefully exploited to better estimate these parameters in practice [Cai et al., 2007].

4.3.5 Modeling Phrases vs. Proximity

This section describes a final simple experiment studying the effect of modeling ordered phrases vs. proximity. While previous work has shown these two distinct types of features provide complementary benefit to retrieval accuracy, we show here that at least in the case of modeling pair-wise sequential dependencies, nearly identical performance can be achieved across collections by modeling proximity only. Specifically, we replace the ordered `#1` Indri operator with the unordered `#uw2` proximal operator and leave other model settings unchanged. Results are shown in Table 4.5.

While proximity is still being matched at two different window sizes, results suggest the ordering-restriction is unnecessary under settings in which the MRF model is typically used in practice. Earlier work on biterm modeling similarly showed small differences in accuracy when employing ordering-restricted and ordering-ambivalent models [Srikanth and Srihari, 2002]. This raises several interesting questions. Do phrasal vs. proximity features really provide distinct value, or are we merely

Feature Used	Robust04	W10g	GOV2
ordered `#1`	25.64	19.14	27.40
unordered `#uw2`	25.61	18.95	27.20

Table 4.5: MAP retrieval accuracy of the sequential-dependency MRF (§2.6) on verbose queries using all topics. The standard MRF feature testing ordering of query term dependencies (`#1`) is seen to have negligible impact vs. order-ambivalent matching (`#uw2`). Usual 85-15-5 component weights, unigram weighting, proximal `#uw8` features, and ML estimation of phrasal and proximal parameters is used.

observing a graduated effect of proximity at different window sizes? Important named-entities and collocations being matched may simply occur rarely enough in reversed order that the unordered feature approximates the ordered feature with reasonable accuracy. Would modeling a broader range of window sizes simultaneously be useful with smaller window size suggesting stronger dependencies? Will the utility of distinctly modeling phrases vs. proximity become more clearly marked as we more fully estimate the MRF model, using longer and non-sequential dependencies and abandoning ML estimation of feature weights? We plan to investigate these and related issues in future work.

4.4 Discussion

We began this chapter by emphasizing the distinction between model and estimation in evaluating a document ranking method's effectiveness. Lexical retrieval models are actually remarkably expressive but have typically not been estimated to their full potential. While recent work in *learning to rank* [Joachims et al., 2007] has demonstrated a variety of new and effective retrieval models, the more sophisticated estimation techniques and additional features that typically go into these new models can alternatively be employed to better estimate existing lexical models and function as a layer atop classic search engines (Ch. 3) [Bendersky and Croft, 2008, Kumaran and Allan, 2007, 2008].

Consider the model and estimation method underlying classic language modeling [Ponte and Croft, 1998] and probabilistic approaches [Sparck Jones et al., 2000]. Both can be viewed as constrained log-linear models adopting a specific feature set and restrictions on parameters. Unigram modeling can be viewed as a log-linear model in which the set of permissible parameterizations Λ is restricted to the probability distribution Θ^Q and the feature set F consists solely of the (log) document model Θ^D:

$$log\ p(Q|D) \propto \Theta^Q \cdot \Theta^D = \Lambda \cdot F$$

Building on the derivation in [Lafferty and Zhai, 2003], we can similarly express the probabilistic approach as:

$$log\ \frac{p(D|Q,r)}{p(D|Q,\bar{r})} = |D|\ \Theta^D \cdot\ log\ \frac{p(w|Q,r)}{p(w|Q,\bar{r})} = F \cdot \Lambda$$

another constrained log-linear model where r and $\bar{r}$ denote relevant and non-relevant term distributions and Λ is estimated differently in the two cases. Historically it has been a point of contention

which of these two models should be preferred [Lavrenko, 2004, Nallapati, 2006]. However, if we accept Lavrenko's argument for dropping $|D|$ feature scaling on the grounds that concatenating a document with itself ought not to double its relevance score [Lavrenko, 2004], both models utilize nearly identical features, differing by only a *log* factor, and are in fact rank-equivalent under equal parameterization. In short, we see the two approaches are constrained not by their models but by their fixed estimation strategies. Less constrained estimation would unlock greater modeling power.

We view the MRF approach (§2.6) as a general linear model which is more expressive than the ways it which has typically been used. We have discussed at length how the MRF has historically assumed one weight parameter per feature class: λ_T, λ_U, and λ_O. While parameter tying within each feature class certainly simplifies estimation, modeling power is reduced, and we have seen how breaking this parameter tying indeed has a positive effect on retrieval accuracy. The MRF variants for *full independence*, *sequential dependence*, and *full dependence* similarly provide a means of enforcing constraints on model sparsity to simplify estimation, but they represent only three fixed options out of an infinite space of possible continuous parameterizations. While it is impractical to model an exponential number of features at retrieval time, off-line methods for feature selection and estimation can be explored and subsequently applied to dynamically select and weight the most important features at run-time. Adopting the general linear model perspective of the model has the further benefit of enabling us to exploit the large body of existing techniques for maximizing such models, including recent work specifically targeting maximization of ranking metrics [Joachims et al., 2007].

4.5 Conclusion

This chapter addressed generalization and better estimation of Metzler and Croft's Markov random field (MRF) (§2.6) approach to document retrieval. While the original MRF method estimated a parameter for each feature class from data, we showed how parameters within each class were implicitly estimated using the same maximum-likelihood assumption employed with the standard unigram. Because this scheme does not model context-sensitivity, its use particularly limits retrieval accuracy with verbose queries in which many terms appearing in the query are not strongly related to the core information need and so ought to be assigned lower weight. By employing supervised estimation instead, however, we showed this deficit could be remedied. Retrieval experiments conducted with verbose queries on three TREC document collections showed our better-estimated MRF consistently out-performs both the baseline MRF and the supervised unigram model. Additional experiments using blind-feedback and evaluation with optimal weighting demonstrate both the immediate value and further potential of performing more accurate MRF model estimation.

Chapter 5

Simpler Unigram Estimation for Verbose Queries

Previous work in IR based on vector similarity has shown the importance of applying inverse document frequency (IDF) in weighting both document and query terms [Salton and Buckley, 1987]; we also found document frequency useful to include in Regression Rank secondary features (§3.2.3). While it has been shown that smoothed estimation in language modeling implicitly captures an IDF-like effect via inverse collection frequency (ICF) [Zhai and Lafferty, 2004], such smoothed estimation is typically employed only on the document side (to infer the latent document model Θ^D). In contrast, the latent information need Θ^Q underlying the query Q is usually estimated without any provision for capturing IDF-like term importance. However, explicit use of ICF in estimating Θ^Q has been shown to yield better search accuracy in practice than achieved with simple maximum-likelihood (ML) estimation [Smucker and Allan, 2006]. In this chapter, we describe a simple method for estimating Θ^Q which leverages both IDF and ICF in combination and which consistently achieves better search accuracy than ICF (and ML). Moreover, we show this simple model achieves comparable search accuracy to more sophisticated learning-based methods like Regression Rank (Ch. 3).

A fundamental principle of the language modeling approach is that more accurate estimation of Θ^Q and Θ^D should lead to improved search accuracy. While our simple method of estimation follows the same as ad hoc approach as the original ICF technique, we discuss how it can be considered a stepping stone toward more principled estimation based on logistic regression. The promise of such an extensible regression-based approach is more accurate inference of Θ^Q by being able to integrate other forms of statistical evidence beyond IDF. While our current model's combination of ICF and IDF remains incredibly naive, in comparison to Regression Rank it is far simpler, and as we shall show, no less effective for searching with verbose queries. Consequently, we suggest new techniques for supporting verbose queries also be compared against this baseline in addition to maximum-likelihood (ML) or ICF to provide more rigorous empirical evaluation.

A related question left unanswered by previous chapters is how keyword and verbose queries

should be supported in tandem? If we assume a user interface like today's ubiquitous search box in which users can submit arbitrary natural language queries, it is important that the system be able to effectively cope with both keyword and verbose input[1]. While one might sidestep this problem by changing the user interface to stipulate use of verbose queries, let us assume for now that we do not wish to be so dependent on the user interface. In this scenario, do we advocate using a single model for both short and verbose queries, or would it be more effective to distinguish between the two types of queries and use a different model in each case?

Previous work has shown ICF weighting performs comparably to ML for short queries [Smucker and Allan, 2006]. With Key Concepts (§2.7.2), simple analysis shows its method for term weighting effectively reverts to ML for most keyword queries. In particular, if a query consists entirely of a single noun phrase or contains no noun phrase at all, conditions which cover most keyword queries by definition, there is no noun phrase to be emphasized over other query terms and all terms are assigned uniform weight in Θ^Q. In this chapter, we evaluate both Regression Rank and Gigaword weighting for keyword search and show both methods perform comparably or nearly comparably to ML. In sum, this indicates that even if we could perfectly distinguish between the two types of input queries and apply the best method in each case, doing so would yield minimal benefit over simply using one of the above verbose query methods for both types of queries. Consequently, we advocate the simpler single-model approach.

The next section describes our simple linear function for combining IDF and ICF to estimate Θ^Q. Following this (§5.2), we evaluate the method on both verbose and keyword queries and compare to previously discussed methods. We then present discussion (§5.3) of further questions and issues raised by our findings and elaborate further on issues mentioned above. Our conclusion summarizes the chapter and its contributions.

5.1 Method

Following our initial work on Regression Rank, we conducted leave-one-out analysis of features using development topics for the same three document collections (Table 2.4) on which the model was originally evaluated. While we had incrementally added and evaluated features during model development, we did not test whether the final feature space could be trimmed back down to achieve a more parsimonious model. Leave-one-out analysis showed that it could in fact be tremendously simplified and still achieve comparable accuracy. Results of this analysis are summarized in Table 5.1. A lesson learned from this experience was the value of performing such analysis early to accurately gauge feature importance, and we elaborate on this point in later discussion (§5.3). In this section, we define the simplified model. Evaluation in §5.2 will compare it empirically to the original version.

As shown in Table 5.1, estimating Θ^Q on the basis of Gigaword [Graff et al., 2005] ICF and IDF values alone performs comparably to use of the full feature set (§3.2.3). While we would have

[1] While the space of possible user queries is clearly richer than the simple keyword/verbose distinction being made here, we view our two-way taxonomy a starting point for subsequent extension to other query categories of interest.

Features	MAP
Baseline ML	19.67
Non-CF/DF	21.37
Robust04 CF/DF	22.65
Robust04 CF/DF + Non-CF/DF	22.87
Gigaword CF/DF	23.44
Robust04 & Gigaword CF/DF	23.66
All	23.80

Table 5.1: Leave-one-out analysis of Regression Rank features as a function of mean-average precision (MAP) achieved on Robust04 using development topics (Table 2.4). Statistical significance is not reported, but CF/DF features are seen to clearly absorb initial improvement shown from non-CF/DF features. Experimental setup follows that used in §3.3.

hoped to have identified additional useful evidence for estimating Θ^Q, it appears what we did learn at minimum was the utility of modeling both IDF and ICF. Note that we will use DF/CF instead of IDF/ICF somewhat arbitrarily since the distinction between them is absorbed once we tune λ_3:

$$ln(tf) + ln(\frac{N}{df}) = ln(tf) - ln(df) + ln(N) = \lambda_1\, ln(tf) + \lambda_2\, ln(df) + \lambda_3$$

where N is number of documents in the collection and we use $tf + 1$ and $df + 1$ to avoid infinities with out-of-vocabulary query terms). As before (§3.2.2), we do not bother normalizing our estimate of Θ^Q to get a proper probability distribution since document ranking is invariant to scaling of Θ^Q.

In addition to achieving equally effective ranking using only these two features, we also observed Regression Rank learned similar weights across document collections for these two features and the bias parameter. This similarity suggested a single setting might work well across all three document collections. To test this, we performed a sweep of around 300 parameterizations in the proximity of these settings and evaluated accuracy on each collection (still using development topics). For both Robust04 and GOV2 document collections (Table 2.4), maximal MAP search accuracy was obtained with parameterization $\lambda_{1:3} = \{0.45, -0.52, 1.0\}$. In the case of W10g, this parameterization achieved 0.9% worse MAP (absolute) than the best parameterization evaluated, but this difference was not statistically significant. Consequently, we accepted the above configuration as a strong single parameterization for all three document collections. Since the parameter optimization above was limited to the approximately 300 settings considered, better parameterizations certainly may exist. However, evaluation below shows this parameterization achieves search accuracy comparable to what Regression Rank achieved with unconstrained estimation on its entire original feature set.

5.2 Evaluation

This section compares search accuracy achieved by simple CF and DF-based estimation of Θ^Q vs. that of ML, Key Concepts (§2.7.2), and Regression Rank (Ch. 3). As in Ch. 4, we also evaluate integration of unigram estimation techniques with Markov Random Field (MRF) modeling (§2.6)

and use of pseudo-relevance feedback (PRF) [Lavrenko and Croft, 2001]. To address the one vs. two model question for supporting keyword and verbose queries in tandem, search accuracy with keyword queries is also reported.

Six variant CF/DF methods are evaluated, all of which estimate Θ^Q (unnormalized) as a linear function of CF and DF statistics: $\Theta^Q = \lambda_1 \, ln(tf) + \lambda_2 \, ln(df) + \lambda_3$. Methods differ by which corpus statistics are drawn from and how the statistics are weighted via $\lambda_{1:3}$:

- IDF methods set $\lambda_{1:3} = \{0, -1, ln(N)\}$, where N is number of documents in the given document collection, to compute standard IDF weighting of $ln\frac{N}{df}$.
- Inverse CF (ICF) methods set $\lambda_{1:3} = \{-1, 0, ln(T)\}$, where T is total number of tokens in the given document collection, to weight query terms by $ln\frac{T}{cf}$ [Smucker and Allan, 2006].
- "CF+DF" methods use $\lambda_{1:3} = \{0.45, -0.52, 1.0\}$ as discussed in the preceding section.

Statistics are alternately collected from the given document collection used for retrieval or from the Gigaword corpus [Graff et al., 2005]. With "CF+DF", the same parameterization tuned for Gigaword is used unchanged for collection-specific CF+DF weighting. Additional improvement from the collection-specific approach may be possible by performing collection-specific tuning.

Evaluation is conducted using the same document collections and topics as in previous chapters (Table 2.4). As before, we use the `description` field of topics as verbose queries and the `title` field as keyword queries, with † and ‡ marking significance ($p < 0.05$) and high significance ($p < 0.01$) respectively according to a randomization test [Smucker et al., 2007].

5.2.1 Verbose queries

Table 5.2 presents search accuracy results for verbose queries without use of feedback, complementing earlier results presented in Tables 4.1 and 4.2. The first row reports the original ML baseline used in both Ch. 3 and Ch. 4. Below this, we see that while Gigaword-IDF/ICF do not consistently improve over the ML baseline, Collection-IDF/ICF do show consistent improvement. Below this, earlier Key Concepts and Regression Rank results are repeated from earlier. Note that in comparison to Collection-IDF, Key Concepts shows no improvement and Regression Rank shows improvement for Robust04 only. Collection CF+DF shows improvement for Robust04 and W10g but performance declines for GOV2, perhaps due to mismatch between the Gigaword-based parameterization and the much larger size of this corpus. In contrast, Gigaword CF+DF shows consistent improvement across all three document collections with respect to both Collection IDF and Collection CF+DF. Consequently, we suggest new techniques for supporting verbose queries also be compared against this baseline in addition to maximum-likelihood (ML) or ICF to provide more rigorous empirical evaluation.

Next we see the results of Metzler and Croft's MRF model using sequential dependency (§2.6), which shows no significant improvement over collection IDF weighting across document collections. The row below this repeats earlier results from Ch. 4 for using Regression Rank to estimate the

Model	Θ^Q Estimation	Robust04 P@5	Robust04 MAP	W10g P@5	W10g MAP	GOV2 P@5	GOV2 MAP
Unigram	Maximum Likelihood	47.64	24.51	39.20	18.61	52.21	25.22
	Gigaword-ICF	47.71	$25.43^{\ddagger}$	40.00	19.55	$55.70^{\ddagger}$	$26.50^{\ddagger}$
	Gigaword-IDF	47.23	25.24	39.00	20.57	$56.51^{\dagger}$	26.32
	Collection-ICF	48.19	$25.95^{\ddagger}$	40.40	$20.25^{\ddagger}$	$56.38^{\ddagger}$	$26.90^{\ddagger}$
	Collection-IDF	48.76	$25.83^{\ddagger}$	40.00	$21.73^{\ddagger}$	$57.05^{\ddagger}$	$26.71^{\ddagger}$
	Key Concepts	47.55	25.91	41.40	20.40	57.05	27.44
	Regression Rank	$52.05^{\ddagger}$	$27.33^{\ddagger}$	40.60	22.01	54.50	27.35
	Collection CF+DF	$50.52^{\ddagger}$	$26.98^{\ddagger}$	40.00	$20.82^{\ddagger}$	54.77	25.65
	Gigaword CF+DF	$51.33^{\ddagger}$	$27.39^{\ddagger}{}_{\ddagger}$	41.40	$21.93^{\ddagger}{}_{\dagger}$	$55.70^{\dagger}$	$27.90^{\ddagger}{}_{\ddagger}$
MRF (§2.6)	Maximum Likelihood	49.32	25.64	38.80	19.14	56.38	27.40
	Regression Rank	$54.30_{\dagger}$	$28.48^{\ddagger}{}_{\ddagger}$	42.00	$23.05^{\ddagger}{}_{\dagger}$	57.18	$29.51^{\ddagger}{}_{\ddagger}$
	Gigaword CF+DF	$52.85^{\ddagger}{}_{\dagger}$	$28.31^{\ddagger}{}_{\ddagger}$	40.60	$22.89^{\ddagger}{}_{\dagger}$	58.39	$29.92^{\ddagger}{}_{\ddagger}$

Table 5.2: Search accuracy in mean-average precision (MAP) and precision of top 5 ranks (P@5) for verbose queries (`description` field) using all topics and collections from Table 2.4. Results complement earlier Tables 4.1 and 4.2. Unigram results: For all three collections, while Gigaword-IDF and Gigaword-ICF do not consistently improve over the ML baseline, Collection-IDF and Collection-ICF [Smucker and Allan, 2006] do show consistent improvement. Subsequent methods are then compared against Collection-IDF: Key Concepts (§2.7.2), Regression Rank (Ch. 3), Collection CF+DF, and Gigaword CF+IDF. Key Concepts shows no significant improvement. Regression Rank improves for Robust04 only. Collection CF+DF improves for Robust04 and W10g MAP but declines for GOV2. Gigaword CF+DF shows consistent improvement across collections. Subscript$_{\dagger}$ indicates statistical significance of Gigaword CF+DF accuracy over Collection CF+DF. MRF results: We start with the standard sequential dependency MRF model (with its default ML unigram estimation) (§2.6). In comparison to Collection-IDF, the MRF shows no statistical improvement. Regression Rank and Gigaword CF+IDF unigram estimation are alternately integrated into the MRF model. Superscript† and subscript$_{\dagger}$ here indicate statistical significance of the combined model vs. the baseline MRF and the given unigram method, respectively. In both cases the combination improves significantly over either component used individually.

Model	Robust04 Test	Robust04 All	W10g Test	W10g All	GOV2 Test	GOV2 All
Key Concepts (§2.7.2)		29.35				
epi-HAL [Hoenkamp et al., 2009]		31.0				
Markov Random Field (§2.6)	38.92	30.09	19.99	20.02	32.37	30.26
Regression Rank	37.03	30.52	21.77	22.48	30.36	28.96
MRF + Regression Rank	$39.13_{\ddagger}$	$31.82^{\ddagger}{}_{\ddagger}$	$23.19^{\dagger}{}_{\dagger}$	$23.05^{\ddagger}$	$32.91_{\ddagger}$	$31.20_{\ddagger}$
Gigaword CF+DF	38.65	30.85	22.00	22.56	31.67	29.77
MRF + Gigaword CF+DF	39.33	$31.67^{\ddagger}{}_{\ddagger}$	$23.64^{\ddagger}{}_{\ddagger}$	$23.41^{\ddagger}{}_{\dagger}$	$33.92^{\dagger}{}_{\ddagger}$	$31.84^{\ddagger}{}_{\ddagger}$

Table 5.3: Search accuracy in mean-average precision (MAP) with pseudo-relevance feedback (PRF) [Lavrenko and Croft, 2001] for verbose queries (`description` field) over all collections and on test vs. all topics from Table 2.4. Results complement those presented earlier in Table 4.3. $\text{Score}^{m}{}_{u}$ superscripts and subscripts are used here to indicate statistical significance of each combined model vs. its individual components: the MRF (m) and the unigram (u). Results show that Gigaword CF+DF performs comparably to Regression Rank both in isolation and in combination with the MRF model. We also compare to results for epi-HAL [Hoenkamp et al., 2009], a technique based on query expansion using the Hyperspace Analog to Language (HAL); its results are reported for Robust04 only. Results for Key Concepts with PRF were generated using non-PRF Indri [Strohman et al., 2004] queries provided by its authors, then applying the same PRF parameterization used with Regression Rank and the MRF; further improvement with Key Concepts can be expected by tuning PRF parameters for it. Statistical significance comparisons with Key Concepts and epi-HAL are not reported.

MRF's unigram component1. At bottom, we similarly use the Gigaword CF+DF model to estimate MRF unigram weights, and just as we saw in isolation, in combination with the MRF model the Gigaword weighting scheme performs comparably to Regression Rank once more. To summarize improvement without PRF in comparison to the ML baseline, Gigaword CF+DF estimation with the MRF model achieves relative MAP improvements of ca. 15-20% (15.5% for Robust04, 23.0% for W10g, and 18.6% for GOV2, comparing top and bottom rows of Table 5.2).

Table 5.3 presents search accuracy results for verbose queries using PRF. These results complement earlier results from Table 4.3. As in §4.3.3, PRF is used only to better estimate the unigram component of the MRF model. The first three rows copy results shown in the earlier table. Row 4 presents accuracy of Gigaword CF+DF weighting with PRF and shows accuracy strictly greater than Regression Rank with PRF in all cases (significance not evaluated). Similarly, row 5 presents results of using Gigaword CF+DF to estimate MRF unigram weights and then applying PRF.

Finally, we describe an additional experiment using a variant form of Key Concepts (§2.7.2). Recall Key Concepts estimates term weights as a mixture between predicted concept weights and the ML estimate of term weights. Given the relative strength of Gigaword CF+DF vs. ML, we tried replacing Key Concepts' ML component with Gigaword CF+DF instead. We then re-tuned the mixture weights for the two components. On Robust04 with development topics, this yielded no MAP improvement over using Gigaword CF+DF alone. Given this negative result, we did not proceed to further test the idea on other document collections. Our conclusion from this experiment is that Key Concepts' method of predicting term weights (in addition to Regression Rank's) is

	System	Year	TREC-7 P@5	TREC-7 MAP	TREC-8 P@5	TREC-8 MAP
	`title` only					
TREC	Okapi `ok7as` (TREC-7)	1998	53.20	26.14		
	Queens `pir9At0` (TREC-8)	1999			51.60	30.63
	`description` only					
TREC	NEC `nectitechdes` (TREC-7)	1998	58.40	25.84		
	UMass `INQ602` (TREC-8)	1999			49.60	24.92
No PRF	Dirichlet Smoothing (§2.4.3)	2001	46.80	17.96	44.80	23.26
	Two-Stage Smoothing (§2.7.2)	2002	41.60	18.10	48.40	23.10
	MRF (§2.6)	2005	48.40	18.95	46.80	23.71
	Collection-ICF [Smucker and Allan]	2006	50.40	20.11	46.00	24.77
	Term Dependent Smoothing [Mei et al.]	2007	44.00	19.60	47.60	24.60
	Key Concepts (§2.7.2)	2008	48.00	20.21	46.00	23.64
	Regression Rank	2009	55.20	21.96	50.40	26.47
	Gigaword CF+DF		54.00	21.74	46.80	26.19
	MRF + Regression Rank	2009	58.40	22.89	58.40	27.14
	MRF + Gigaword CF+DF		54.80	22.36	50.40	26.46
With PRF	Dirichlet Smoothing (§2.4.3)	2001	46.80	23.12	52.00	27.11
	MRF (§2.6)	2005	50.80	23.85	53.20	28.34
	Collection-ICF [Smucker and Allan]	2006	49.20	24.79	50.80	28.09
	Key Concepts (§2.7.2)	2008	48.80	24.41	50.80	27.28
	Regression Rank	2009	55.60	25.66	54.80	29.25
	MRF + Regression Rank	2009	57.20	26.28	56.80	29.93
	MRF + Gigaword CF+DF		56.40	26.25	55.20	29.78

Table 5.4: Search accuracy in mean-average precision (MAP) and precision of top 5 ranks (P@5) for verbose queries (`description` field) on the Robust04 collection (Table 2.4) using topics from TREC-7 (351-400) and TREC-8 (401-450). Note that Regression Rank and Gigaword CF+DF were tuned on a superset of these topics (351-450) whereas official TREC results reflect blind evaluation. Statistical significance is not reported. Results here extend earlier Table 2.5.

not learning useful information beyond the CF and DF statistics being leveraged here. This is also consistent with an earlier experiment not reported in which a similar combination between Regression Rank and Key Concepts yielded no improvement over the individual models.

Tables 5.4 and 5.5 provide additional comparison between Regression Rank and Gigaword CF+DF methods, as well as comparison to previous work. Overall, results presented in this section paint a convincing argument that the simple Gigaword CF+DF approach achieves comparable search accuracy on verbose queries as Key Concepts or Regression Rank, at least with regard to the original setup proposed for each of those methods. We consider further ramifications of these results for those methods in the discussion section (§5.3).

ID	Old Topic Set				New Topic Set				Hard Topic Set				Combined Topic Set			
	MAP	P10	%no	area	MAP	P10	%no	area	MAP	P10	%no	area	MAP	P10	%no	area
t	.317	.505	5	.033	.401	.545	6	.089	.183	.374	12	.016	.333	.513	5	.038
d	.315	.507	8	.023	.407	.547	2	.074	.162	.382	12	.013	.334	.515	7	.028
Without PRF																
1	.2318	.4025	11.5	.0098	.2993	.4673	4.1	.0426	.0988	.2560	20.0	.0054	.2451	.4153	10.0	.0118
2	.2413	.4200	10.5	.0122	.3180	.4735	4.1	.0540	.1096	.2920	14.0	.0064	.2564	.4305	9.2	.0149
3	.2482	.4085	13.5	.0084	.3058	.4653	4.1	.0378	.1092	.2700	20.0	.0055	.2595	.4197	11.6	.0101
4	.2456	.4015	13.5	.0078	.3141	.4776	4.1	.0360	.1057	.2440	22.0	.0022	.2591	.4165	11.6	.0100
5	.2629	.4370	9.0	.0134	.3158	.4735	8.2	.0347	.1278	.3040	16.0	.0080	.2733	.4442	8.8	.0151
6	.2627	.4370	10.0	.0121	.3196	.4898	4.1	.0393	.1205	.2760	16.0	.0066	.2739	.4474	8.8	.0141
7	.2737	.4565	7.5	.0158	.3302	.4959	6.1	.0408	.1337	.3140	12.0	.0092	.2848	.4643	7.2	.0182
8	.2709	.4565	8.0	.0146	.3330	.5041	4.1	.0434	.1263	.3060	12.0	.0077	.2831	.4659	7.2	.0169
With PRF [Lavrenko and Croft, 2001]																
1	.2660	.4315	16.5	.0065	.3770	.4939	6.1	.0678	.1157	.2820	26.0	.0019	.2879	.4438	14.5	.0089
2	.2792	.4485	16.5	.0072	.3897	.5082	6.1	.0786	.1309	.3060	24.0	.0024	.3009	.4602	14.5	.0104
3	.2799	.4380	15.5	.0069	.3725	.5000	8.2	.0567	.1240	.2860	24.0	.0039	.2981	.4502	14.1	.0088
4	.2749	.4240	17.0	.0046	.3694	.4857	12.2	.0344	.1145	.2620	30.0	.0011	.2935	.4361	16.1	.0060
5	.2907	.4670	12.0	.0100	.3645	.5061	10.2	.0398	.1300	.3140	16.0	.0067	.3052	.4747	11.6	.0118
6	.2904	.4585	14.0	.0091	.3820	.5204	8.2	.0550	.1230	.2760	22.0	.0035	.3084	.4707	12.9	.0115
7	.3029	.4915	10.0	.0125	.3809	.5061	8.2	.0446	.1386	.3440	12.0	.0076	.3182	.4944	9.6	.0150
8	.2994	.4760	11.5	.0120	.3873	.5020	6.1	.0563	.1314	.3160	22.0	.0053	.3167	.4811	10.4	.0152

Best TREC Robust04 `title` (t) and `description` (d) runs
t. `pircRB04t3`
d. `pircRB04d4`

Other Methods
1. Dirichlet Smoothing [Lafferty and Zhai, 2001]
2. MRF (§2.6)
3. Collection-ICF [Smucker and Allan, 2006]
4. Key Concepts (§2.7.2)
5. Regression Rank (Ch. 3)
6. Gigaword CF+DF
7. MRF + Regression Rank
8. MRF + Gigaword CF+DF

PRF [Lavrenko and Croft, 2001] results were generated with the following Indri [Strohman et al., 2004] settings:

1. `fbDocs` = 10 (fixed)
2. `fbTerms` = 50 (fixed)
3. `fbMu` = 0 (default)
4. `fbOrigWeight` = 0.4 (tuned for MRF on topics 301-450)

Table 5.5: Comparison of methods on the TREC 2004 Robust track (see earlier Table 2.6). Evaluation is performed on the Robust04 document collection (Table 2.4) using four topic sets defined by the track: "old" (301-450, 601-650), "new" (651-700), "hard" (50 topics from 301-450 identified in the Robust03 track overview), and "combined" (all 250 topics). Note that new topics reflect blind evaluation while other topics do not. Besides usual metrics of mean-average precision (MAP) and precision-at-10 (P10), two non-standard metrics are reported which focus on difficult topics: "%no", referring to the percent of topics for which P10 = 0, and "area", referring to area under the MAP curve for the worst quarter topics. The latter two metrics were computed via a publicly available NIST script used in the original tracks: `http://trec.nist.gov/data/robust/robust2004_eval.pl`.

Query Length	Query Count	ML		Regression Rank		Oracle	
		P@5	MAP	P@5	MAP	P@5	MAP
1	1	60.00	22.32	-	-	-	-
2	32	52.12	28.96	53.75	28.70	56.88	30.39
3	42	60.48	31.05	61.43	31.43	64.76	35.80
4	21	59.05	31.79	56.19	31.44	72.38	41.00
5	2	40.00	16.31	50.00	16.42	40.00	21.92
all	98	57.35	30.14	57.55	30.14	63.27	34.73

Table 5.6: P@5 and MAP search accuracy for 98 keyword queries (`title` field) on the GOV2 collection, broken down by query length. By definition, single-term queries assign all probability mass in Θ^Q to the term and so achieve identical ranking under all estimation methods. Statistical significance is not reported, but ML and Regression Rank are clearly seen to perform comparably.

Θ^Q Estimation	Robust04		W10g		GOV2	
	P@5	MAP	P@5	MAP	P@5	MAP
Maximum Likelihood	48.11	25.32	32.16	19.49	56.62	29.61
Gigaword CF+DF	48.67	25.57	32.80	19.94	54.63	$28.68^{\dagger}$

Table 5.7: Search accuracy in mean-average precision (MAP) and precision of top 5 ranks (P@5) for keyword queries (`title` field) using all topics and collections from Table 2.4. The only statistically significant difference between ML vs. Gigaword CF+DF is observed for GOV2 MAP ($p = 0.0256$).

5.2.2 Keyword Queries

As mentioned in the introduction, a question left unanswered in previous chapters was whether the same method of estimating Θ^Q should be used for both keyword and verbose queries. Assuming the search engine's interface allows users free expression in formulating their queries, the engine must support a variety of input queries. With regard to the ML baseline, we have suggested earlier that ML makes more sense for keyword than verbose queries because keywords tend to be more carefully selected and thereby more uniformly important than are terms in verbose queries. While one could imagine that keyword queries also stand to benefit from better weighting, differences would likely be more subtle as a consequence of this fact. Moreover, by virtue of being shorter, keyword queries employ a smaller vocabulary (i.e. feature space) providing a corresponding smaller potential for improvement. Can our methods for supporting verbose queries improve upon ML weighting with keyword queries, or at least match it, or do they under-perform ML in this case?

Previous work has already shown ICF weighting achieves comparable accuracy as ML for title queries [Smucker and Allan, 2006], and our analysis in the introduction shows clearly how the Key Concepts (§2.7.2) method of term weighting effectively reverts to ML with title queries. Consequently, we focus here on evaluating accuracy of Regression Rank (Ch. 3) and Gigaword CF+DF weighting methods.

Table 5.6 reports P@5 and MAP search accuracy on the GOV2 collection for 98 `title` queries (a subset of topics 701-850 excluding official topics used in the TREC 2008 Relevance Feedback track; see §6.2.3 for additional details). Regression Rank is compared to the ML baseline and to "oracle"

1/0 weighting of query terms (i.e. as reported earlier for verbose queries in Table 3.2). Results show that about 4.5% absolute MAP improvement is possible under oracle weighting, roughly half of that seen earlier with verbose queries. In terms of accuracy achievable in practice, we see that Regression Rank breaks even with respect to the ML baseline, showing it could be used as a single model for both keyword and verbose queries, at least for GOV2.

Table 5.7 compares ML and Gigaword-based estimation of Θ^Q for keyword queries on all three document collections and topics, reporting P@5 and MAP search accuracy. While Regression Rank was seen to show no difference vs. ML in Table 5.6, here we do see Gigaword CF+DF showing a statistically significant decrease vs. ML on MAP accuracy for the GOV2 collection (0.93% absolute, $p = 0.0256$). Nonetheless, given the relatively small size of this difference, it seems safe to regard these methods for verbose queries also as largely comparable to ML for keyword queries. Consequently, we believe it is reasonable to use a single one for both verbose and keyword queries.

Of course it is still possible that another distinction between query types could motivate use of multiple estimation strategies. For example, Figure 3.1 showed that like any technique, Regression Rank performed worse on some verbose queries than the ML baseline, and likely a per-query analysis of keyword queries would show something similar. If we could somehow effectively distinguish not between our current categories of keywords vs. verbose queries, but instead between queries where one estimation technique does better than the other, then by definition we could improve overall search accuracy. The challenge, of course, would be recognizing what distinguishes these two classes, implementing an accurate classifier to make this distinction in practice, and verifying this trend generalized beyond the queries considered here. We leave such an investigation to future work.

5.3 Discussion

IDF weighting has a long history and has significantly influenced all of the major retrieval paradigms [Jones, 2004]. With vector similarity, for example, it has been seen that the best weighting scheme incorporates an IDF weighting factor for both document and query terms [Salton and Buckley, 1987], but more convincing theoretical justification for why IDF should be applied on both document and query side (i.e. effectively squared) has been needed. Fortunately, theoretical justification for smoothed estimation (i.e. frequentist regularization or Bayesian modeling of priors) is well-established in statistics for accurately inferring a latent distribution given finite observable evidence. This is significant because Zhai and Lafferty showed typical smoothing of the document unigram Θ^D via collection statistics implicitly applies ICF weighting with its IDF-like effect [Zhai and Lafferty, 2004]. This suggested we might interpret IDF's effectiveness as arising from similar benefit as smoothed estimation, and that further benefit may be achievable in principled fashion by focusing effort on better estimation of Θ^D and Θ^Q.

While such smoothed estimation is routinely employed for estimating the document unigram Θ^D, ML estimation of the query unigram Θ^Q is still typical and thus misses out on the IDF-like weighting of query terms seen to be effective with vector similarity. While various forms of query expansion

and rewriting are commonly employed, they do not address this missing IDF effect. Moreover, we cannot simply estimate Θ^Q like Θ^D to achieve the same implicit ICF weighting effect in a principled manner. While ICF estimation does directly capture this missing IDF effect and has been shown to be empirically effective [Smucker and Allan, 2006], it lacks theoretical justification. Achieving this ICF effect in a theoretically-driven manner remains an open problem.

In this work we have directly evaluated explicit use of DF in combination with CF for language modeling and shown that this combination improves over use of IDF/ICF alone. Thus our work complements the previous work on ICF weighting [Smucker and Allan, 2006]. While simply plugging CF and DF into a linear function is an admittedly ad hoc way to go about estimation, a minor change to the function to perform logistic regression would put us on well-trodden ground for using arbitrary statistics to model an observed probability distribution. While this still would not provide a generative story for the distribution, it does let us apply rich estimation methodology in modeling it. This was the intuition underlying Regression Rank and allowed us to explore a variety of features in addition to CF and DF for modeling Θ^Q. As discussed above, what we are really not after is a way of modeling IDF, but rather accurate estimation of Θ^Q which IDF weighting merely approximates.

Our evaluation considered four variant CF/DF methods: Collection-IDF, Collection CF+DF, Gigaword-IDF, and Gigaword CF+DF. While Collection-IDF was seen to outperform Gigaword-IDF, Gigaword CF+IDF outperformed Collection CF+DF. How might this effect be explained? We typically expect to see collection-specific statistics be more informative and larger corpora provide more reliable statistics (Gigaword has 1.8B tokens; see Table 2.4 for collection statistics). One might also expect web statistics to be more noisy than those gathered from newswire content. While Gigaword and Robust04 are both newswire content and Gigaword is far larger, we nonetheless saw that Robust04 IDF statistics were more useful than Gigaword IDF statistics for Robust04 retrieval. So it would appear that having collection-specific statistics is most significant, except we then observe Gigaword CF+DF outperform Collection CF+DF. Our explanation at present is that Collection CF+DF should perform better but does not due to lack of collection-specific tuning. The drop in accuracy on GOV2 is particularly disturbing and seems the strongest indicator of this given the difference in collection size vs. Gigaword. This issue needs to be further investigated. To the extent Gigaword CF+DF weighting is more effective than collection-specific weighting, we would like to further see whether additional improvement is possible by using larger corpora, or in the other direction, if accuracy falls when using less robust statistics from smaller sample of the corpus. We would also like to try leveraging Gigaword or other external corpora in conjunction with collection-specific statistics. We have tried something similar elsewhere with some success (Ch. 7).

What do the results presented here imply for the supervision-based approaches described earlier? In terms of practical effectiveness today, the methods described in this chapter are simpler, more efficient, and equally accurate, and so clearly preferable. However, thinking beyond today to the IR systems of tomorrow, we still believe the future lies in exploring richer features beyond CF and DF. Learning-based frameworks provide us with an excellent vehicle for conducting such research by facilitating a division of concerns between feature design and estimation. Key Concepts (§2.7.2) and

Regression Rank both providing extensible frameworks for exploring new features, and both stand to benefit as more effective estimation techniques are developed. Moreover, as query logs continue to provide an ever-growing source of implicit relevance judgments, there is a tremendous opportunity for learning-based models to exploit such logs for lifetime learning, allowing search accuracy to continually improve without additional human effort as more training data becomes available.

A final lesson of this chapter was seeing that feature weights learned during estimation only coarsely indicate the relative importance of model features, especially when using a non-sparse prior like L2 (§3.2.4). In comparison, leave-one-out experiments provide a far more accurate indication of relative feature importance. Hence we advocate performing such leave-one-out experiments for analyzing the relative contribution of features rather than inspecting learned feature weights.

5.4 Conclusion

We have shown that estimating Θ^Q using a simple combination of CF and DF statistics consistently achieves better search accuracy than both ICF [Smucker and Allan, 2006] and ML baselines. Moreover, this simple scheme achieves accuracy comparable or better to more sophisticated learning-based approaches (Ch. 3, [Bendersky and Croft, 2008]). Consequently, we suggest new techniques for supporting verbose queries be compared against this method in addition to maximum-likelihood (ML) to provide a more rigorous empirical evaluation.

Search accuracy was also evaluated for keyword queries. We saw Regression Rank matched search accuracy of ML for GOV2, while Gigaword CF+DF achieved roughly comparable performance to ML across document collections (with slightly lower MAP on GOV2). Given this, along with previous ICF results [Smucker and Allan, 2006] and our analysis of Key Concepts' behavior on keyword queries, it seems reasonable to use existing models for verbose queries to support keyword search as well. This provides a simple means of achieving strong performance on both without having to distinguish between query types.

Our closing discussion highlighted several points. While IDF weighting been known to be useful for weighting both queries and documents, language modeling approaches have typically only captured this effect on the document side, which has reduced effectiveness with verbose queries. ICF weighting [Smucker and Allan, 2006] provided an effective but ad hoc way to perform IDF-like weighting on the query side, and our combination of CF and DF represents a more effective extension to this ad hoc technique. As shown by earlier oracle experiments (Ch. 3), more effective estimation techniques have the potential to improve search accuracy significantly beyond today's accuracy levels. While the linear functions used here are ad hoc, they are not far from more principled methods for logistic regression which allow arbitrary evidence to be used in modeling an observed probability distribution. This is the spirit of what Regression Rank intended to capture via its secondary features (§3.2.3), the potential still worth chasing after even if the initial attempt showed only the importance of modeling CF in combination with DF. Consequently, we still believe learning-based

approaches present the most promising direction for future research. Such frameworks enable exploration of novel features beyond CF and DF, richer retrieval models than simple bag-of-words, and stand to empirically benefit as better estimation techniques are developed and more training data becomes available.

Chapter 6

Integrating Relevance & Pseudo-relevance Feedback

Previous chapters have focused on the distinction between short and verbose queries, suggesting that while verbose queries often better characterize the information being sought, models must accurately infer the relative importance of the various details present to effectively incorporate supporting details without losing sight of the core request. In this chapter, we turn our attention to another form of useful verbosity in characterizing information needs: feedback documents. When a system knows or believes it has identified one or more documents exemplifying the information being sought, once more there is a tremendous opportunity to better model the information need if the system can effectively distinguish between relevant and non-relevant information contained in those example documents. As with longer queries, however, feedback documents are verbose, and not all the information expressed in them is equally important or relevant to a given information need. Consequently, we see once more that effective estimation is paramount.

While we have focused primarily on ad hoc retrieval (§2.2) in earlier chapters, we now turn our attention primarily to the RF task. Basic methodology of relevance and pseudo-relevance feedback was introduced earlier (§2.5). In this chapter, we first elaborate on the close relationship between verbose queries and use of feedback documents in terms of enlarging the feature space of terms for which we have observational evidence supporting estimation. Following this, we describe in §6.2 an estimation strategy which combines RF, PRF, and Markov Random Field (MRF) modeling.

6.1 Relationship with Verbose Queries

Just as longer queries tend to be more informative in describing information needs than keyword queries, feedback documents provide additional context for interpreting the user's information need. Recognition of this close connection motivates our work with feedback documents, and we proceed now to describe this connection slightly more formally.

IR models have traditionally distinguished between relevant and non-relevant documents on the basis of words, meaning their feature space is defined by the vocabulary employed. Let V denote a random variable over possible vocabularies. Maximally $V = v^C$, where C is the collection of documents to be ranked and v^C denotes all terms occurring in the collection[1]. Assuming documents are ranked by simple query-likelihood (§2.4.3), we have $V = v^Q$, i.e. we assign non-zero weights only to terms observed in the query Q. Note that this vocabulary restriction is purely an estimation issue; while the retrieval model itself is well-defined over the entire vocabulary v^C, lacking prior knowledge our observation of Q provides our only basis for estimating term weights[2]. By this same token, however, longer queries will tend to introduce new terms and thereby enable us to leverage a larger vocabulary in ranking. Relevance feedback provides an even better foothold for estimation, expanding the vocabulary to include related terms found in feedback documents in addition to the query. Finally, applying pseudo-relevance feedback with collection documents can potentially expand the vocabulary up to the maximum v^C. To summarize, both verbose queries and feedback documents are seen to provide the same basic advantage over short queries: they enable more fine-grained discrimination between documents by enlarging the representational feature space (for which we have observational evidence) via additional terms. What sets them apart is predominantly a question of scale since relevance feedback provides evidence for estimating more terms.

6.2 Integrating RF and PRF with MRF modeling

As a means of effectively performing such estimation, we describe in this section an approach combining relevance feedback, pseudo-relevance feedback, and Markov random field modeling of term interaction. Overall effectiveness of our combined model and the relative contribution from each component is evaluated on the GOV2 webpage collection. Given 0-5 feedback documents, we find each component contributes unique value to the overall ensemble, achieving significant improvement individually and in combination. Comparative evaluation in the 2008 TREC Relevance Feedback track further shows our complete system typically performs as well or better than peer systems.

6.2.1 Introduction

We present here a strategy for effectively leveraging varying amounts of explicit feedback (documents): none (a.k.a. *ad hoc* retrieval), one, a few, or many. This is combined with use of PRF to automatically induce additional feedback documents to further expand the query [Lavrenko and Croft, 2001, Zhai and Lafferty, 2001]. Although PRF has been primarily investigated with ad hoc retrieval, it has the potential for great effectiveness in the RF setting as well since explicit feedback improves system ranking for automatically identifying related documents. Alongside PRF, we also

[1] We assume out-of-vocabulary (OOV) query terms not found in any document are ignored in retrieval.

[2] Given some prior distribution over terms, e.g. from collection statistics or earlier queries, smoothing to terms unseen in the query is certainly possible. However, since query processing time is directly proportional to query length, benefits of such broad smoothing must be balanced against processing time.

investigate the benefit of modeling term interactions in the RF scenario. Specifically, we adopt Metzler and Croft's Markov random field (MRF) modeling of sequential dependencies between terms (§2.6).

We evaluate the benefit from applying each of these techniques individually and in combination. Given 0-5 feedback documents, we find each component contributes unique value to the overall ensemble, achieving significant improvement individually and in combination. Additional experiments using RF in absence of MRF or PRF yield results consistent with community wisdom that a little feedback can make a big difference. Finally, we describe comparative evaluation of our complete system in the 2008 TREC Relevance Feedback track.

6.2.2 Method

This section describes our overall approach. Our approach is based in the query-likelihood paradigm for information retrieval (§2.4.3), and we adopt the aforementioned MRF model (§2.6) in particular to capture interactions between pairs of adjacent query terms. As in Ch. 4, we replace the MRF's default maximum-likelihood estimation of the unigram with a more effective strategy. Whereas we employed supervised unigram estimation in that case, here we employed relevance and pseudo-relevance feedback.

Given an input query Q and feedback documents F, our approach may be summarized as follows:

1. A unigram document model Θ^D is estimated for each document $D \in F$ via Dirichlet smoothing (§2.4.3)
2. A unigram query model Θ^Q is estimated from Q via maximum-likelihood[3] (Equation 2.4.17).
3. A unigram RF model Θ^F is estimated as the average document model over the set of positive (i.e. relevant) feedback documents (Equation 2.5.1)
4. An improved unigram query model $\Theta^{Q'}$ is produced by linearly mixing Θ^Q and Θ^F models (Equation 2.5.3)
5. $\Theta^{Q'}$ is used as the unigram component f_T in the MRF model to yield $P'_\Lambda(D|Q)$ (Equation 6.2.1)
6. A unigram pseudo-relevance model Θ^P is estimated based on $P'_\Lambda(D|Q)$ (Equation 2.5.4)
7. The PRF unigram likelihood $\Theta^P \cdot \Theta^D$ is linearly mixed with the $P'_\Lambda(D|Q)$ MRF model (Equation 6.2.2)

Note that unigram likelihood (Equation 2.4.17) can be equivalently formulated as an MRF in which $\lambda_T = 1$ and $\lambda_O = \lambda_U = 0$. This means an improved unigram model $\Theta^{Q'}$ (e.g. better estimated via feedback) can be used in place of the MRF's standard f_T unigram model:

$$P'_\Lambda(D, Q) \propto \lambda_T[\Theta^{Q'} \cdot log\, \Theta^D] + \lambda_O f_O + \lambda_U f_U \tag{6.2.1}$$

[3]Since we are operating on keyword rather than verbose queries, we do not apply Regression Rank or Gigaword-based estimation of query terms since results presented earlier found limited benefit from doing so (§5.2.2).

When using PRF in conjunction with the MRF model, we must specify how Θ^P is mixed with original model: query model mixing (i.e. in the f_T component) or ranking function mixing. We adopt Indri's formulation [Metzler et al., 2006] incorporating PRF at the level of the ranking function:

$$P''_\Lambda(D|Q) = \lambda_P[\, log\, \Theta^P \cdot \Theta^D] + (1-\lambda_P) P'_\Lambda(D|Q) \tag{6.2.2}$$

using $P'_\Lambda(D|Q)$ as defined in Equation 6.2.1. Note PRF is limited here to unigram modeling; we do not estimate dependency statistics from PRF for revising f_O and f_U components since previous work has shown little benefit from doing so [Metzler and Croft, 2007a].

6.2.3 Evaluation

This section describes evaluation performed in developing and testing our model. Table 6.1 provides a complete listing of all model parameters and identifies which remain fixed in our experiments. We follow previous work in setting MRF proximity parameters for window size $w_{proximity}$ and Dirichlet smoothing $\mu_{proximity}$.

Track Protocol and Metrics

Model evaluation was performed as part of our participation in the 2008 TREC Relevance Feedback Track. A goal of the track was to establish strong baselines for current RF techniques under varying amounts of explicit feedback:

A: no feedback (i.e. ad hoc retrieval)

B: 1 relevant document

C: 3 relevant and 3 non-relevant documents

D: 10 judged documents

E: large amounts of feedback (40-800 documents)

Each feedback set was included as a subset of its larger successors. Retrieval experiments were conducted on the GOV2 webpage collection (25,205,179 documents) with 264 title-field queries drawn from topics of 2004-2006 Terabyte tracks (TREC topics 701-850) and the 2007 Million Query track (50 and 214 topics, respectively). Documents chosen for feedback achieved the highest median retrieval ranks in the earlier track from which the topic was drawn using the best run submitted by participating groups. All odd-numbered and some even-number Terabyte topics were excluded from the test set and so available for model development; evaluation on test topics was blind. Top-2500 document rankings were submitted for official runs though reported results include top-1000 ranked documents only.

Cumulative metric performance across topics is generally computed by a simple (arithmetic) average over per-query metric performance. The one exception, geometric-mean average precision (`gMAP`), adopts the geometric mean instead in order to focus metric attention on difficult topics.

Component	Parameter	Value
Unigram	μ	1700
Relevance Feedback	λ_F	varied
	k_F	varied
MRF	λ_T	varied
	λ_O	varied
	λ_U	$1-\lambda_T-\lambda_O$
	$w_{proximity}$	8
	$\mu_{proximity}$	4000
Pseudo-rel Feedback	λ_P	varied
	k_P	50
	$\|\mathcal{P}\|$	10

Table 6.1: Parameters of our combined model.

Primary metrics used were (arithmetic) mean-average precision (`MAP`) and top-10 precision (`P@10`), as reported by `trec_eval` 8.1[4]. Besides `gMAP`, we also report R-Precision (`rprec`): precision after R documents retrieved, where R is the number of relevant documents for each topic. Results are marked as significant† ($p < 0.05$), highly significant‡($p < 0.01$), or neither according a non-parametric randomization test computed by Indri's `ireval` [Smucker et al., 2007].

Experimental Setup

Indri [Strohman et al., 2004] formed the basis of our retrieval model. Since Indri does not provide a facility for performing RF, however, we estimated the feedback model Θ^F externally. Queries were stopped at query time using a 418 word INQUERY stop list [Allan et al., 2000] and then Porter stemmed[5]. Recall that term pair features f_O and f_U from the dependency model (Equation 2.6.4) correspond to co-occurrence statistics tracking pairs of words occurring consecutively or within some proximity of one another. It is worth noting that Indri replaces stopwords with out-of-vocabulary tokens and so use of stopwords does not affect distance between terms in computed co-occurrence statistics.

For model development, track protocol did not specify which documents to use for feedback with non-test topics. While it would have been ideal to choose documents achieving high rank under ad hoc retrieval, mirroring testing conditions, we simply took feedback documents for each topic according to their order in the collection assessments. Initially we tried evaluating cross-validated performance over different choices of feedback documents, but we ended up abandoning this practice due to time constraints. Since our RF method made no use of negative-feedback, our choice of feedback involved only relevant documents. For condition D, we always used 5 relevant documents rather than vary the number per topic as in testing conditions. Finally, with condition E we simply used all relevant documents under an assumption that once so many feedback documents

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

[5] `http://www.tartarus.org/martin/PorterStemmer`

Model	**A**	**B**	**C**	D
Unigram	29.18	‡30.84	†31.94	‡33.49
PRF		32.50‡	32.47	‡34.32‡
MRF	32.04‡	32.55†	‡34.61‡	‡35.62‡
MRF+PRF	35.28‡	34.78‡	35.37†	‡36.66‡

Table 6.2: (Mean) average precision achieved by different model configurations on development topics. Parameterization is consistent with Table 6.3 except $k_F = 150$ is used with all feedback runs. Statistical significance is reported by prefix † and ‡ comparing against cell to left (i.e. less feedback), while suffix compares PRF & Unigram, MRF & Unigram, and MRF+PRF & MRF.

were available, the exact number would make little difference. We did not test this assumption, however, and so it bears some scrutiny in future work.

Tuning was performed with feedback documents included in the evaluation in accordance with a reading of track protocol which later proved to be mistaken. This led to selection of parameter settings which likely overfit feedback. Despite the non-optimality of this tuning process, our development set results presented below do properly exclude feedback documents and so support useful analysis. Of the 98 topics originally used in tuning, we discard three which have fewer than five non-feedback relevant documents, leaving 95 for evaluation. Since condition E tuning used all relevant documents as feedback, its performance can only be evaluated with feedback documents included. Consequently, this condition is largely omitted in our discussion of development set results.

Results on Development Topics

Parameter values were tuned on development topics via grid search (cf. [Salton and Buckley, 1987]), resulting in the values listed in Table 6.3. Results in Table 6.2 compare baseline unigram `MAP` with that achieved using PRF, MRF, and MRF+PRF combined. While results generally show improvement with increasing feedback, the more interesting observation is seeing how the techniques contribute and interact with one another in comparison to the baseline and across feedback conditions. With the sole exception of PRF in condition C, we see PRF and MRF modeling each yield significant improvement over the baseline across feedback conditions with MRF seen to be the stronger of the two. Furthermore, the MRF+PRF combination achieves additional significant improvement over MRF modeling alone. With condition E (not shown), neither PRF or the MRF model improved over the baseline. However, this result is inconclusive since condition E development set results could not be evaluated without retrieved feedback documents.

We submitted nine runs for official evaluation: five unigram runs with no PRF (conditions A-E) and four MRF+PRF runs (conditions A-D). No MRF+PRF run was submitted for condition E since we did not observe improvement from either technique on this condition while tuning. Evaluation of these runs on development topics is shown in Table 6.4. Results show fairly steady improvement for unigram runs but a more complicated picture for MRF+PRF runs. While `gMAP, rprec`, and `P@10` steadily improve with increasing feedback, `map` is flat for A-C. However, both `map` and `P@10`

Model	Run	k_F	λ_F	λ_T	λ_O	λ_P
Unigram	A2	-	-	-	-	-
	B2	250	0.3	-	-	-
	C2	150	0.45	-	-	-
	D2	150	0.45	-	-	-
	E1	250	0.8	-	-	-
MRF+PRF	A1	-	-	0.8	0.1	0.5
	B1	150	0.3	0.8	0.1	0.75
	C1	150	0.45	0.9	0.05	0.85
	D1	150	0.45	0.9	0.05	0.85

Table 6.3: Parameterization of submitted runs. MRF+PRF values are identical for C and D conditions.

Model	Run	MAP	gMAP	rprec	P@10
Unigram	A2	29.18	21.65	35.27	54.32
	B2	‡30.84	24.22	36.52	†57.89
	C2	†31.94	26.27	38.14	57.37
	D2	‡33.49	27.89	39.15	‡62.42
MRF+PRF	A1	35.28‡	26.42	38.62	60.53‡
	B1	34.78‡	28.33	39.50	61.68†
	C1	35.37‡	29.88	40.15	61.89†
	D1	†36.66‡	31.42	40.88	†64.95

Table 6.4: Unigram and MRF+PRF results on development topics. Statistical significance is reported for `map` and `P@10` (only) by prefix † and ‡ comparing against cell above (i.e. less feedback) while suffix compares Unigram vs. MRF+PRF runs using comparable feedback.

show significant improvement for condition D.

Results on Test Topics

Official test set results of our nine submitted runs are presented in Table 6.5. `MAP, gMAP, rprec`, and `P@10` metrics are computed on top-1000 retrieved documents with relevance determined by NIST pooling assessment of 31 Terabyte track topics. The pool consisted of the top-10 ranked documents from each run submitted by a participant. `MTC` corresponds to Carterette et al.'s Minimal Test Collections evaluation algorithm [Carterette et al., 2006] and `statAP` comes from Aslam and Pavlu's statistical MAP method [Aslam et al., 2006]; both algorithms were used in the TREC Million-query Track. Million-query track runs also contributed to the pools.

Unigram results demonstrate a steady improvement in retrieval accuracy across all but `gMAP` metrics with growing amounts of feedback. The largest MAP improvement is seen moving to condition E's large amount of feedback (4.11% absolute over condition D). A slightly smaller MAP improvement is seen as we go from ad hoc retrieval (condition A) to condition B's having a single relevant document: 3.66% (absolute). Similar trending is observed with high-rank P@10 retrieval: 11.61% and 5.49%, respectively (absolute). Regarding `gMAP`, it would seem topic drift caused by

Model	Run	MAP	gMAP	rprec	P@10	MTC	statAP
Unigram	A2	13.43	4.05	16.48	24.19	4.90	22.91
	B2	‡17.09	6.99	21.09	†29.68	6.22	29.07
	C2	‡19.50	8.66	22.66	32.58	7.03	32.27
	D2	20.64	9.29	23.67	†36.45	7.06	32.16
	E1	†24.75	14.85	27.35	‡48.06	7.32	35.00
MRF+PRF	A1	21.46‡	11.43	25.15	32.90	5.64	27.99
	B1	20.96	11.63	23.56	33.87	6.04	29.59
	C1	†22.96†	13.68	25.75	37.74	7.01	33.87
	D1	†24.29†	14.93	27.42	40.65	7.03	32.16

Table 6.5: Official results of our runs on test topics. Run name indicates feedback condition and run ID. Runs are divided between unigram results (no PRF) and results using both sequential dependency (§2.6) and PRF. Statistical significance is reported for `map` and `P@10` (only) following the same conventions used in Table 6.4. While the general ranking is consistent between MAP and statAP, note that the former is based on shallow pooling; see track overview for details [Buckley and Robertson, 2009].

	MAP		P@10	
System	A-E	B-E	A-E	B-E
Brown	22.89	23.23	38.64	40.08
uogRF09	22.08	22.68	38.64	38.87
UAmsR08PD	19.22	20.09	35.17†	36.78†
UIUC	18.55†	20.09†	32.52†	35.41‡
FubRF08	17.85†	19.58†	32.26†	35.48‡

Table 6.6: Relative performance achieved by five of the top systems participating in the track, as measured by simply averaging official test topic MAP and P@10 accuracies across the various feedback conditions. As mentioned in Table 6.5, note reported MAP scores are based on shallow pooling. Column "A-E" averages over all conditions, while "B-E" compares feedback conditions only (no ad hoc "A"). Statistical significance measured by a two-tailed paired t-test is reported for low significance† ($p < .05$) and high significance‡ ($p < .01$). Refer to track overview [Buckley and Robertson, 2009] and official track results for more detailed comparison.

feedback is seen to hurt performance, though this loss diminishes as greater feedback reduces drift. However, note a very different trend is observed on development topics (Table 6.4). It may be this difference in trends is simply a byproduct of differences between how feedback documents were selected for development and test sets. On the other hand, since official evaluation only included top-10 ranked documents in pooling, assessment may have been biased in favor of easier topics for which many relevant documents would be seen early in the ranked list. Finally, since we use identical system configurations for conditions C and D (which provide comparable feedback), we expected their results should be quite similar, and `MTC` and `statAP` metrics bear this out.

MRF+PRF results are less clear in that condition B results decline in comparison to ad hoc retrieval under `MAP` and `rprec` metrics while improving under all other metrics. This drop is likely due to overfitting. Otherwise similar trends are observed: we see improvement with increasing feedback. C and D conditions again appear roughly comparable, with D generally performing slightly better except in the case of `statAP`. Overall, we see that depending on the base model used, ca. 15-85% relative improvement in MAP accuracy is achieved via document feedback (84.3% for the unigram, 13.2% for the MRF+PRF, comparing top and bottom rows for each in Table 6.5).

Table 6.6 shows the relative strength of our overall system in comparison to four other competitive submissions to the 2008 TREC Relevance Feedback track. Performance is summarized by simply averaging official MAP and P@10 accuracies across the various feedback conditions. Results shown our system typically performed as well or better than peer systems. The track overview [Buckley and Robertson, 2009] and official track results provide more thorough details for comparison.

6.3 Conclusion

Verbose queries and feedback documents both offer an opportunity to better infer a query's latent information need in comparison to short queries *if* we can effectively infer the relative importance and salience of additional terms. Both enlarge the representational feature space of terms for distinguishing between relevant and non-relevant documents and provide observational evidence for estimating term importance over this enlarged vocabulary. As such, we saw estimation is again a key issue as in earlier chapters.

As a specific contribution, we described an effective strategy for combining relevance feedback, pseudo-relevance feedback, and Markov random field modeling techniques for document retrieval. Using a large web collection, we evaluated an overall combination strategy while assessing the contribution from each component in presence of the others. Given 0-5 feedback documents, we found each component contributed unique value to the overall ensemble, achieving significant improvement individually and in combination.

Comparative evaluation in the 2008 TREC Relevance Feedback track further showed our system typically performed as well or better than peer systems. Use of proximity (e.g. features in our MRF model) and/or PRF was generally seen to help in combination with RF across participating systems that employed one or the other. Use of negative feedback (e.g. via Rocchio) generally provided

little benefit. Interestingly, all of the competitive participants' systems displayed some form on non-monotonicity in accuracy with increasing feedback. While we identified problems with overfitting in our system, as discussed earlier, it remains to be seen this is explanation is sufficient in general.

While our approach to RF here was limited to unigram feedback, an interesting topic for future work will be exploring term dependency selection from feedback documents for incorporation into f_O and f_U MRF components (Equation 2.6.4). Previous work has shown little benefit from PRF dependency modeling [Metzler and Croft, 2007a], but RF dependency modeling may prove to be more helpful. We would also like to explore use of RF in conjunction with supervised unigram modeling such as described in Ch. 3.

Another interesting direction to explore would be applying the Gigaword-based weighting strategy presented in Ch. 5 with the expanded queries created by feedback. Because such queries are far longer than the typical question and sentence verbose queries considered in Chapters 3-5, maximum-likelihood / relative frequency-based estimation has more robust statistics to work with and so may prove less problematic than with the more natural verbose queries considered earlier. Nonetheless, it would be interesting to test this hypothesis experimentally.

Chapter 7

Dirichlet-smoothed Bigram Modeling and Collection Expansion

While previous chapters have described estimation methods for better retrieving text documents, this chapter investigates better estimation methods in the context of retrieving spontaneous speech documents. As in Ch. 5, we evaluate system accuracy for both keyword and more verbose queries[1]. As with the earlier text retrieval experiments, we find here that retrieval accuracy of spontaneous speech documents can also be significantly improved by better estimation, and we investigate bigram modeling [Song and Croft, 1999] as an incremental improvement over the traditional bag-of-words representation considered in Ch. 3 and an alternative to Markov Random Field modeling (Chapters 4 and 5).

In particular, this chapter describes two simple but effective smoothing techniques for the standard language model (LM) approach to information retrieval. First, we extend the popular unigram Dirichlet smoothing technique (§2.4.3) to bigram modeling. Second, we propose a method of *collection expansion* for more robust estimation of the LM prior, particularly intended for sparse collections. Retrieval experiments on the MALACH archive [Oard et al., 2004] of automatically transcribed and manually summarized spontaneous speech interviews demonstrates strong overall system performance and the relative contribution of our extensions.

7.1 Introduction

In the language model (LM) paradigm for information retrieval (IR), a document's relevance is estimated as the probability of observing the query string as a random sample from the document's underlying LM (§2.4.3). The standard unigram LM approach has been shown to have a strong theoretical connection [Zhai and Lafferty, 2004] to classic TF-IDF statistics and comparable empirical

[1] In this chapter we consider verbose queries as the concatenation of `title` and`description` fields; evaluation of `description` field queries alone remains for future work.

performance [Fang et al., 2004] to other state-of-the-art approaches like vector similarity (§2.4.1) and the "probabilistic" approach (§2.4.2). This chapter presents two modest smoothing-based extensions in the LM paradigm.

Whereas the unigram model and other standard approaches to retrieval typically assume bag-of-words independence between terms, modeling even a simple notion of term dependency represents a useful step toward richer modeling of queries and documents. Previous work in bigram modeling provided a valuable first step in this direction within the LM paradigm and demonstrated its empirical merit [Song and Croft, 1999]. Subsequent to this, Dirichlet smoothing with unigram models was found to elegantly and effectively capture the intuition that longer documents should require less smoothing since they provide more support for the maximum-likelihood (ML) estimate [Zhai and Lafferty, 2004]. While one would expect bigram models could similarly benefit, we have not seen a Dirichlet-smoothed bigram model described or evaluated in the IR literature. Consequently, we describe such a model here and report on its effectiveness. As with the earlier bigram formulation [Song and Croft, 1999], our approach easily generalizes to higher-order mixtures.

The second extension we describe addresses smoothing at the collection-level. As suggested above, smoothing plays an important role in inferring accurate document LMs, and it can be accomplished in a principled manner via *maximum a posteriori* estimation using a prior model. For IR, the prior is typically estimated from collection statistics, but just as estimating a robust document model is often challenging due to document sparsity, estimating the prior from a small (i.e. sparse) collection can be equally problematic. To address this, we propose estimating the prior from an "expanded" version of the collection containing additional statistics drawn from external corpora. This idea closely parallels previous work expanding documents with similar ones found in external sources [Singhal and Pereira, 1999]. Previous work in topic detection and tracking has also leveraged external corpora to gain more robust statistics when only few documents have been seen [Allan et al., 1998]. Here, collection-wide statistics are expanded via external corpora to enable more robust estimation of the LM prior. We show simple collection expansion via broad, external corpora significantly improves retrieval accuracy.

We evaluated our model and extensions via retrieval experiments on the MALACH archive of automatically transcribed and manually summarized spontaneous speech interviews [Oard et al., 2004]. These experiments were conducted as part of the Cross-Language Speech Retrieval track's shared task [Pecina et al., 2008] at the 2007 Cross Language Evaluation Forum. Results show the overall competitive performance of our system as well as the relative contribution of our extensions.

The remainder of the chapter is presented as follows: methodology is discussed in §7.2, relevant details of the MALACH collection and pre-processing are described in §7.3, evaluation procedure and results are presented in §7.4, and §6.3 summarizes and describes future work.

7.2 Method

7.2.1 Dirichlet-smoothed Bigram Modeling

We adopt the Dirichlet-smoothed unigram formulation presented earlier (§2.4.3) and seek to extend this formulation to bigram modeling. To accomplish this, we similarly smooth the empirical bigram estimate with hyper-parameter μ_1 pseudo-counts distributed fractionally according to the collection prior bigram model, $P(w_i|w_{i-1}, C)$:

$$P(w_i|w_{i-1}, D, C) = \frac{f_{w_{i-1},w_i} + \mu_1 P(w_i|w_{i-1}, C)}{f_{w_{i-1}} + \mu_1} \tag{7.2.1}$$

Unigram and bigram models can then be easily mixed by treating our smoothed unigram distribution $P(w|D, C)$ as an additional prior on the bigram model and adding in μ_2 pseudo-counts drawn from it:

$$P(w_i|w_{i-1}, D, C) = \frac{f_{w_{i-1},w_i} + \mu_1 P(w_i|w_{i-1}, C) + \mu_2 P(w|D, C)}{f_{w_{i-1}} + \mu_1 + \mu_2} \tag{7.2.2}$$

Whereas earlier work inferred the hyper-parameters α in Equation 2.4.13 from data in order to realize a coupled prior tying unigram and bigram models [MacKay and Peto, 1995], our formulation can be viewed as a less sophisticated alternative that reduces α to three hyper-parameters, μ, μ_1, and μ_2, to be tuned on development data.

7.2.2 Collection Expansion

The second extension we describe addresses more robust estimation of the LM prior by performing smoothing at the collection-level. As discussed above, ML estimation of document LMs is hurt by document sparsity, and hence *maximum a posteriori* estimation is commonly employed instead using an informative prior induced from the collection. The effectiveness of this strategy, however, relies on accurate estimation of the prior, which can be challenging for small (i.e. sparse) collections.

To address this, we propose estimating the prior from an "expanded" version of the collection containing additional data drawn from external corpora. This approach parallels traditional work in document expansion in which collection documents are expanded with external, related documents [Singhal and Pereira, 1999]. In both cases, the underlying idea of expansion being employed is characteristic of a broad finding in the learning community that having additional similar data enables more robust estimation. In our case of *collection expansion*, we hope to compensate for collection sparsity by drawing upon "similar" data from external corpora.

For this work, we simply leveraged two broad English newspaper corpora: the Wall Street Journal (WSJ) and the North American News Corpus (NANC) [Graff, 1995]. Specifically, we expanded the collection as a linear mixture with 40K sentences (830K words) from WSJ (as found in the Penn Treebank [M. Marcus et al., 1993]) and 450K sentences (9.5M words) from NANC, with tunable hyper-parameters specifying integer mixing ratios between corpora. The particular corpora and mixing scheme used could likely be improved by a more sophisticated strategy. For example, results in §7.4 show significant improvement for modeling manually-written summaries but not for

automatic transcriptions, likely due to mismatch between the external corpora and the automatic transcriptions. Bigram statistics in expansion corpora were not collected across sentence boundaries, which were manually annotated in WSJ and automatically detected in NANC [McClosky et al., 2006].

7.3 Data

This section describes the retrieval collection used and pre-processing performed. A more complete description of the collection can be found elsewhere [Oard et al., 2004, 2006, Pecina et al., 2008].

Data used came from the Survivors of the Shoah Visual History Foundation (VHF) archive of interviews with Holocaust survivors, rescuers, and witnesses. A subset of this archive was manually and automatically processed by VHF and members of the MALACH initiative (Multilingual Access to Large Spoken Archives) in order to improve access to this archive and other such collections of spontaneous speech content. As part of this effort, interviews were manually segmented and summarized, as well as automatically transcribed (several variant transcriptions were produced). Manual transcription was limited and not provided for interviews included in the retrieval collection. Each interview segment was also manually assigned a set of keywords according to a careful ontology developed by VHF, and two versions of automatically detected keywords were also provided. Topics used for retrieval were based on actual information requests received by VHF from interested parties and were expressed in typical TREC-style with increasingly detailed title, description, and narrative fields [Oard et al., 2004].

In terms of pre-processing, sentence boundaries were automatically detected to collect more accurate bigram statistics. Boundaries for manual summaries were detected using a standard tool [Reynar and Ratnaparkhi, 1997] and interview segment keyword phrases were each treated as separate sentences. We noted the presence of multiple contiguous spaces in automatic transcriptions appeared to correlate with sentence-like units (SUs) [LDC, 2004] and so segmented sentences based on them[2]. Use of automatic SU-boundary detection is left for future work [Roark et al., 2006].

7.4 Evaluation

This section describes system evaluation, including experimental framework, parameter settings, and results. Retrieval experiments were performed as part of the 2007 Cross Language Evaluation Forum's Cross-Language Speech Retrieval (CL-SR) task [Pecina et al., 2008].

We used 25 topics for development and 33 for final testing (the 2005 and 2006 CL-SR evaluation sets, respectively; the 2006 test set was re-used for the 2007 evaluation). For the "manual" retrieval condition, segments consisted of manual summaries and keywords. For the "automatic" condition, we used the ASR2006B transcripts and both versions of automatic keywords. Following previous work [Zhai and Lafferty, 2004], the unigram Dirichlet smoothing parameter μ was fixed at 2000 for

[2]Collection documentation does not discuss this.

Collection	Queries	Dev	CL-SR'05	Test	CL-SR'06	CL-SR'07
Manual	TDN	.3829	-	.2870	.2902	.2847
	TD	.3443	.3129	.2366+	.2710	.2761+
	T	.3161	-	.2348	.2489	-
Auto	TDN	.1623	.2176	.0910	.0768	-
	TD	.1397	.1653	.0785-	.0754	.0855-

Table 7.1: Mean-average precision retrieval accuracy of submitted runs. CL-SR columns indicate representative strong results achieved in that year's track on the same query set [Oard et al., 2006, Pecina et al., 2008]. Runs marked with `+/-` were reported in the 2007 track report to represent statistical significance and non-significance, respectively.

Model	T	TD	TDN
Unigram baseline	.2605	.2722	.2810
Dirichlet bigram	.2545 (-2.3%)	.2852 (4.8%)	.2967 (5.6%)
Collection Expansion	.2716 (4.3%)	.3021 (11.0%)	.3236 (15.2%)
Combination	.2721 (4.5%)	.3091 (13.6%)	.3369 (19.9%)

Table 7.2: Relative improvement in mean-average precision on the development set over the unigram baseline model for Dirichlet-smoothed bigram modeling and collection expansions, alone and in combination (manual condition, no pseudo-relevance feedback).

both manual and automatic conditions. Best performance was usually observed with μ_1 set to 1, while optimal μ_2 settings varied.

A limited pseudo-relevance feedback (PRF) scheme was also employed. As in standard practice, documents were ranked by the model according to the original query, with the most likely documents taken to comprise its feedback set (the number of feedback documents used varied). The query was then reformulated by adding the 50 most frequent bigrams from each feedback document. A tuning parameter specified a multiplier for the original query counts to provide a means of weighting the original query relative to the feedback set. This scheme likely could be improved by separate treatment for unigram feedback and weighting feedback documents by document likelihood under the original query.

Results in Table 7.1 show performance of our five official runs on development and test sets[3]; queries used were: title-only (T), title and description (TD), and title, description, and narrative (TDN). Representative strong results achieved in 2007's and previous years' CL-SR tracks [Oard et al., 2006, Pecina et al., 2008] are also shown, though it should be noted that our results on the development set correspond to tuning on those queries whereas the CL-SR'05 official results do not. Retrieval accuracy was measured using mean-average precision reported by `trec_eval` version 8.1[4].

[3]Following submission of official runs, we found a bug affecting our parsing of the *narrative* field of three test queries. Table 7.1 show system performance with the bug fixed. Without the fix, `Manual-TDN` on the test set was .2577 and `Auto-TDN` was .0831.

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

Table 7.2 shows the impact of our extensions compared to the baseline Dirichlet-smoothed unigram retrieval model for the no-PRF "manual" condition. Of the two extensions, collection expansion is seen to have greater effect, with the combination yielding the best result. The effect of the extensions with the "automatic" condition was marginal (the best absolute improvement seen was 0.3% achieved by the bigram model). With collection expansion, we suspect this is due to the mismatch between the collection's spontaneous speech and the text corpora used for expansion (§7.2), and we plan to investigate use of better matched corpora in future work. As for the bigram model, automatic transcription noise is more problematic than with unigrams since recognition error further impacts prediction of subsequent terms. One strategy for addressing this would be to work off the recognition lattice instead of the one-best transcription. Another challenge to the bigram model is the presence of disfluency in spontaneous speech, which disrupts bigram statistics. Automatic detection and deletion of disfluency could help address this and thereby also render the spoken document more amenable to smoothing via external text corpora [Lease et al., 2006].

For manual retrieval with PRF, the combination of extensions was used in selecting the set of documents for feedback. For PRF runs using this feedback set, the extensions were seen to provide minimal further benefit, with PRF tuning parameters dominating the variance in performance observed. Since PRF produces a query more tailored to collection statistics, expanded collection statistics may be less useful in PRF settings.

7.5 Conclusion

This chapter presented two smoothing-based extensions to the standard language model approach to information retrieval: Dirichlet-smoothed bigram modeling and collection expansion. While we are certainly not the first to suggest bigram modeling for IR (cf. [Song and Croft, 1999]), the formulation we describe is the first we know which combines bigram modeling with Dirichlet-smoothing and demonstrates its effectiveness. Similarly, while there has been previous work in expanding documents with similar ones found in external sources [Singhal and Pereira, 1999], we are not familiar with any previous work in expanding collection-wide statistics via external corpora to enable more robust estimation of the language modeling prior. Results of this latter technique showed clear benefit and suggest its general applicability whenever collections are small, such as with personal or community archives rather than massive Web-scale or corporate collections.

Chapter 8

Future Work

There is always more to do, and this dissertation is no exception to the rule. While we have provided some insights and effective strategies for better modeling natural language in the context of IR (particularly with regard to improving support for verbose queries), many interesting and important questions and issues remain open for further consideration and exploration. As such, this chapter highlights and discusses a few of these topics.

8.1 Abandoning stoplists

The idea of not indexing certain terms via stoplists was introduced in §2.3, and we applied such stopping in all of the presented experiments. However, as mentioned in §2.3, we would like to reiterate that stopping should be applied with caution due to its negative impact on system robustness. Whenever one makes any such *a priori* bet that some particular input will never be observed or be important, Murphy's Law tells us we will almost certainly lose this bet. There is always some query for which every term is important, and stopping will reduce system accuracy on these queries. As a simple example, one might want to find a particular quotation or song lyric using or named by a stopword. We should not ignore a class of legitimate user queries just because we do not have TREC queries exemplifying the phenomenon. As web search has become ubiquitous and search engines collect long logs of input queries, we have increasingly realized the tremendous variety of queries and "long tail" of queries which individually occur infrequently but as a class constitute an important portion of web search traffic to support. In light of such increased awareness, as well as other issues we mention below, we recommend the IR community should discontinue use of stoplists. Some already have (cf. [Fang et al., 2004, Mei et al., 2007, Zhai and Lafferty, 2002]). We expand on this argument below as well as its impact on future research both generally and in the specific case of this dissertation.

When we first began our research looking at verbose queries and studying how query verbosity hurt retrieval accuracy in comparison to keyword search, one of the first things we considered was the "trivial" solution: we inspected some TREC `description` queries and the INQUERY stoplist

(§2.3) to see whether and by what degree accuracy might be improved by simply producing a better stoplist. It quickly because clear, however, that the terms *already* present in the stoplist would be problematic for various queries we could imagine, and while we might better optimize the stoplist for some particular set of verbose queries, adding further terms would only further reduce robustness. Moreover, it seemed existing terms present in the stoplist might already reflect optimization for some benchmark set of known queries. To the extent the IR community has been data-limited in terms of having such benchmark queries for empirical evaluation, systems employing stoplists run the risk of having overfit benchmark queries. In a more realistic usage scenario with many more varied queries or searching a particular domain, stoplists tuned on TREC queries would likely require some form of modification. Popular accuracy metrics favoring average-case performance may also have underestimated the detrimental effect of stoplists in terms of lowered robustness. With practical use of a deployed IR system, even rare failures can evoke strong negative reactions from users. While there has been some attention in recent TREC Robust tracks focusing more on difficult queries, concern has largely addressed issues other than loss in robustness due to stopping, likely due to the small set of queries considered. However, one positive advance from the TREC Robust track was the introduction of evaluating using a geometric rather than arithmetic mean over average precision (gMAP vs. MAP) to emphasize poorly performing topics (§6.2).

More sophisticated approaches to stopping are certainly conceivable. One could perform case-folding subsequent to stopping in order to prevent the particular problem mentioned that could prevent some proper nouns from being indexed. Context could be further used to help disambiguate different senses of polysemous terms. However, assuming OOV terms are indeed ignored, any such static, index-time vocabulary reduction will suffer from the same critical flaw of throwing out terms which will likely be important in some query.

Stopping has other drawbacks as well. While there has been much work on creating sophisticated, elegant, and effective stochastic models for IR, stopping has persisted as something of a bandaid or crutch masking certain errors and preventing them from being addressed in a more principled fashion inside of the system's core formalism. This makes it harder to understand the behavior and effectiveness of such a new system without the crutch so that its limitations could be more easily perceived and directly addressed. Another drawback is that a variety of stoplists have percolated into common use, and this variety serves to further complicate comparison across systems and adds another experimental variable that distracts both analysis and production of IR research away from more critical issues, retarding progress. While we initially adopted Kumaran and Allan's simple 20-word stoplist (§2.7.2) to compare against their work under the same experimental conditions, we subsequently shifted to using the INQUERY stoplist to reproduce experimental conditions of Bendersky and Croft [Bendersky and Croft, 2008] for comparison to their work. INQUERY's more aggressive stopping slightly raised accuracy of the baseline IR system though the change was likely not significant (about 0.5% MAP absolute). But we note again this merely reflected retrieval accuracy on a small query set using metrics focused on average rather than worst case accuracy. It would have been far better to compare all systems without the crutch.

To some degree we can see in stopping a microcosm of tradeoffs between rule-based and stochastic approaches to building intelligent systems. The traditional paradigm of hand-crafting expert systems dominated research in artificial intelligence prior to its statistical revolution but often required a great deal of manual effort, did not generalize well, and suffered from poor robustness. Statistical systems have fared better in these respects, and as an alternative to stopping, recent work has investigated strategies for dynamically determining the relative importance of query terms as part of the core model [Bendersky and Croft, 2008, Kumaran and Allan, 2007, 2008] (Ch. 3). While such strategies do incur the storage and efficiency cost of having to index all document terms and process all query terms, they generalize the idea of stopping to allow more flexibly and robustly modeling relative term importance. This means these dynamic strategies have the potential to improve both average and worst-case IR system accuracy.

In principle, statistical methods for dynamic term weighting or selection should have abandoned stopping entirely, but in practice they have tended to continue using stopping and presumably benefited from doing so. This can likely be attributed to some of the issues raised above: tuned stoplists are both readily available and tend to improve accuracy on existing datasets for standard evaluation metrics, and their use is widely accepted in peer review and a canonical preprocessing step in evaluating a new system. It is harder to make the argument against stopping and justify lower results rather than simply following standard practice. Such criticisms can be fairly leveled at the empirical evaluations presented in this dissertation, and they are openly acknowledged with the hope of bringing more awareness and attention to the issue.

With regard to this dissertation, future work should specifically re-run presented experiments without use of stopping, compare differences, and investigate principled strategies for addressing any loss in accuracy without resorting to the crutch of stopping. More generally, it would likely be useful to assemble a set of queries for which traditional stopwords are important, provide relevance annotations for some document collection, and promote use of gMAP for evaluating performance on this query set. We would like to see stopping become the exception rather than rule, with justification expected for its use in the particular circumstances being presented.

8.2 Query Reduction

Earlier we described previous work in query reduction (§2.7.2). In this section we discuss possibilities for further exploring this approach to supporting verbose queries.

Why remove terms when we can more flexibly weight them? As mentioned earlier, if we want to leverage user interaction, query reduction suggests a fairly simple and intuitive model of interaction [Kumaran and Allan, 2007, 2008]. A different motivation for query reduction would be to support natural queries via a thin client layer built atop a "black box" search engine, e.g. a client providing search capabilities using an external vendor's search engine. For example, Yahoo! enables external developers to develop custom search solutions for different applications and environments

by utilizing its BOSS API[1] for core search technology. Since this API does not allow term weighting, at least at present, one could instead model a mapping from input natural queries to more effective reduced queries with the underlying search engine and generate the latter via automation or interaction. While one could also consider a middle ground between weighting and reduction in which limited integer term weights were expressed via relative frequency (i.e. repeating terms), the underlying search engine may perform poorly on such queries with many term repetitions assuming the engine has been optimized for typical (short) user queries.

While large search accuracy gains have been demonstrated via user interaction [Kumaran and Allan, 2007, 2008], no effective fully-automated system had been reported until recently [Kumaran and Carvalho, 2009]. This motivated our initial work with verbose queries: achieving effective fully-automatic query reduction. We began by reproducing Kumaran and Allan's results for oracle query reduction [Kumaran and Allan, 2007] demonstrating the potential of the approach (see below). Our strategy was to use term weights predicted by Regression Rank (Ch. 3) to rank terms, then predict the number of terms to keep/discard for each input query. On development topics with the Robust04 collection (§3.3), we were able to achieve up to 2% MAP improvement absolute over baseline search accuracy with natural queries and ML estimation by simply picking a fixed length cutoff for all queries in the range of 3-5 terms. We further saw with either oracle term ranking and fixed length cutoff or with predicted term ranking and oracle length cutoff, around 5% MAP improvement was possible. Oracle ranking with oracle length-selection per query could achieve 10% MAP improvement, the very oracle query reduction results we mentioned above. We also performed some simple experiments evaluating prediction of reduction length as a simple fraction of original query length. With the same collection and topics, we found the optimal reduction ratio to be 0.4265 of original query length with fairly small variance, and while this matched the optimal length for 97 of the 149 queries, overall MAP was nevertheless worse vs. using a simple fixed maximum length. Finally, we also tried applying a simple term weight threshold for choosing terms to keep, but again the simple fixed maximum length cutoff performed better.

While these pilot experiments showed promise, like others [Cao et al., 2008] we ultimately found term weighting to be more effective with our system than term selection and so abandoned the latter. Nevertheless, the original motivations for query reduction rather than term weighting still apply, and so it seems worthwhile to comment on how this strategy might be further explored. We discuss ideas below in order of increasing sophistication and computational complexity.

The simplest approach is via traditional stoplists, but as we've argued elsewhere (§8.1), this strategy seems too naive and limiting to be effective. Next, one could consider classifying terms as to whether or not they should be retained. This is similar to the Key Concepts work (§2.7.2) except classification would be performed over terms instead of noun phrases, and the classifier would actually be used for classification instead of term weighting. Also similar is our strategy of independently predicting term weights and, as mentioned above, using a decision threshold on predicted term weight to decide which terms to keep. A simple naive Bayes classifier could estimate unigrams for

[1] http://developer.yahoo.com/search/boss

the relevant and non-relevant distributions ($P(w|R)$ and $P(w|\bar{R})$, respectively) and keep only those terms more probable under the former distribution. The effectiveness of such an approach would depend on how these distributions were estimated and be limited by the bag-of-words assumption.

More interesting models would consider history or relationships between terms. For example, we could model a process of sequential term generation conditioned on the set of terms generated thus far and deciding when to stop generating query terms. We could use features similar to Regression Rank for capturing the importance of terms, and Regression Rank's predicted term weights could themselves be used as well: e.g. predicted weight of possible next terms and their difference vs. that of the previously generated term, aggregate predicted weights over terms generated thus far (i.e. is the reduction "good" enough), etc. Simple features like the original query length could be used in conjunction with more sophisticated query prediction measures [Cronen-Townsend et al., 2002] (i.e. would adding an additional term disrupt the coherency of the returned document set), etc. Use of such prediction measures has now been evaluated in [Kumaran and Carvalho, 2009]. The most complex solution would be to generate all possible reductions and then score them. While there are an exponential number of such reductions to consider in general, we can reduce complexity by considering only those reductions to a fixed length, yielding a polynomial number of reductions to consider. While this last approach would be most general and avoid search errors possible with the generative model, it is the most computationally complex since all candidate reductions must be enumerated and scored.

Bibliography

J. Allan, R. Papka, and V. Lavrenko. On-line New Event Detection and Tracking. In *Proceedings of the 21st international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45, 1998.

J. Allan, M. Connell, W. Croft, F. Feng, D. Fisher, and X. Li. INQUERY and TREC-9. In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 551–562, 2000.

T. G. Armstrong, A. Moffat, W. Webber, and J. Zobel. Has adhoc retrieval improved since 1994? In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 692–693, 2009.

J. Aslam, V. Pavlu, and E. Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 541–548, 2006.

K. Balog, W. Weerkamp, and M. de Rijke. A few examples go a long way: constructing query models from elaborate query formulations. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 371–378, 2008.

M. Bendersky and W. Croft. Discovering key concepts in verbose queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 491–498, 2008.

T. Brants and A. Franz. *Web 1T 5-gram v1, LDC Catalog No. LDC2006T13*. Linguistic Data Consortium, 2006.

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

C. Buckley and D. Harman. Reliable information access final workshop report. *ARDA Northeast Regional Research Center Technical Report*, 2004.

C. Buckley and S. Robertson. Relevance Feedback Track Overview: TREC 2008. In *Proceedings of the Seventeenth Text Retrieval Conference (TREC 2008)*, 2009.

K. Cai, C. Chen, K. Liu, J. Bu, and P. Huang. MRF based approach for sentence retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 795–796, 2007.

G. Cao, J. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250, 2008.

B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275, 2006.

C. Clarke, G. Cormack, and E. Tudhope. Relevance ranking for one to three term queries. *Information Processing and Management*, 36(2):291–311, 2000.

S. Cronen-Townsend, Y. Zhou, and W. Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, 2002.

F. Diaz and D. Metzler. Improving the estimation of relevance models using large external corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, 2006.

W. Fan, M. Luo, L. Wang, W. Xi, and E. Fox. Tuning before feedback: combining ranking discovery and blind feedback for robust retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 138–145, 2004.

H. Fang, T. Tao, and C. Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2004.

N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information systems*, 9(3):223–245, 1991.

R. Gaizauskas, M. Hepple, and M. Greenwood. Information Retrieval for Question Answering: a SIGIR 2004 Workshop. In *SIGIR workshop on information retrieval for question answering*, 2004.

J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, 2004.

X. Geng, T. Liu, T. Qin, A. Arnold, H. Li, and H. Shum. Query dependent ranking using k-nearest neighbor. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, 2008.

D. Graff. *North American News Text Corpus, LDC95T21.* Linguistic Data Consortium, 1995.

D. Graff, J. Kong, K. Chen, and K. Maeda. *English Gigaword, LDC Catalog No. LDC2005T12.* Linguistic Data Consortium, 2005.

E. Hoenkamp, P. Bruza, Q. Huang, and D. Song. An effective approach to verbose queries using a limited dependencies language model. In *Proceedings of the 2nd International Conference on the Theory of Information Retrieval (ICTIR)*, 2009.

B. Hui. Applying NLP to IR: Why and how. Technical report, Department of Computer Science, University of Waterloo, April 1998.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.

T. Joachims, H. Li, T.-Y. Liu, and C. Zhai. Learning to rank for information retrieval (LR4IR 2007). *SIGIR Forum*, 41(2):58–62, 2007.

K. Jones. IDF term weighting and IR research lessons. *Journal of Documentation*, 60:521–523, 2004.

J. Kamps, S. Geva, and A. Trotman. Report on the SIGIR 2008 workshop on focused retrieval. *SIGIR Forum*, 42(2):59–65, 2008.

G. Kumaran and J. Allan. A Case for Shorter Queries, and Helping Users Create Them. In *Proceedings of the 2007 joint conference of Human Language Technologies and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 220–227, 2007.

G. Kumaran and J. Allan. Effective and efficient user interaction for long queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 11–18, 2008.

G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 564–571, 2009.

K. Kwok, L. Grunfeld, H. Sun, P. Deng, and N. Dinstl. TREC2004 robust track experiments using PIRCS. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*, 2005.

J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 111–119, 2001.

J. Lafferty and C. Zhai. Probabilistic Relevance Models Based on Document and Query Generation. In *Workshop on Language Modeling for Information Retrieval*, Pittsburgh, Pennsylvania, USA, May 31–June 1 2003.

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

V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, 2001.

LDC. Simple metadata annotation specification version 6.2. Technical report, Linguistic Data Consortium, 2004. http://www.ldc.upenn.edu/Projects/MDE.

M. Lease, M. Johnson, and E. Charniak. Recognizing disfluencies in conversational speech. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1566–1573, September 2006.

D. D. Lewis and K. Sparck Jones. Natural language processing for information retrieval. *Communications of the ACM*, 39(1):92–101, 1996.

J. Lin. The role of information retrieval in answering complex questions. In *Proceedings of the 2006 joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL)*, pages 523–530, 2006.

M. Marcus et al. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

D. J. C. MacKay and L. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1995.

C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

M. Maron and J. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.

D. McClosky, E. Charniak, and M. Johnson. Effective self-training for parsing. In *Proceedings of the 2006 joint conference of Human Language Technologies and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 152–159, 2006.

Q. Mei, H. Fang, and C. Zhai. A study of Poisson query generation model for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 319–326, 2007.

D. Metzler. *Beyond bags of words: effectively modeling dependence and features in information retrieval*. PhD thesis, University of Massachusetts Amherst, 2007.

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

D. Metzler and W. Croft. Latent concept expansion using markov random fields. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–318, 2007a.

D. Metzler and W. B. Croft. Linear feature-based models for information retrieval. *Information Retrieval*, 10(3):257–274, 2007b.

D. Metzler, T. Strohman, Y. Zhou, and W. Croft. Indri at TREC 2005: Terabyte Track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2006.

G. Mishne and M. de Rijke. Boosting web retrieval through query operations. In *Proceedings of the 27th European Conference on Information Retrieval (ECIR)*, 2005.

S. Na, I. Kang, Y. Lee, and J. Lee. Applying Completely-Arbitrary Passage for Pseudo-Relevance Feedback in Language Modeling Approach. *Lecture Notes in Computer Science*, 4993:626–631, 2008.

R. Nallapati. *The Smoothed Dirichlet Distribution: Understanding Cross-Entropy Ranking in Information Retrieval*. PhD thesis, University of Massachusetts Amherst, 2006.

D. W. Oard, D. Soergel, D. Doermann, X. Huang, G. C. Murray, J. Wang, B. Ramabhadran, M. Franz, S. Gustman, J. Mayfield, L. Kharevych, and S. Strassel. Building an information retrieval test collection for spontaneous conversational speech. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, 2004.

D. W. Oard, J. Wang, G. J. F. Jones, R. W. White, P. Pecina, D. Soergel, X. Huang, and I. Shafran. Overview of the CLEF-2006 cross-language speech retrieval track. In *Working Notes for the Cross Language Evaluation Forum 2006 Workshop*, 2006.

P. Pecina, P. Hoffmannova, G. J. Jones, Y. Zhang, and D. W. Oard. Overview of the CLEF-2007 cross language speech retrieval track. In *Evaluation of Multilingual and Multi-modal Information Retrieval - Eighth Workshop of the Cross-Language Evaluation Forum*. Springer, 2008.

J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, 1998.

M. Porter. The Porter Stemming Algorithm. http://www.tartarus.org/martin/PorterStemmer.

J. C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied Natural Language Processing*, pages 16–19, 1997.

M. Richardson, A. Prakash, and E. Brill. Beyond PageRank: machine learning for static ranking. In *Proceedings of the 15th international conference on World Wide Web (WWW)*, pages 707–715, 2006.

B. Roark, Y. Liu, M. Harper, R. Stewart, M. Lease, M. Snover, I. Shafran, B. Dorr, J. Hale, A. Krasnyanskaya, and L. Yung. Reranking for sentence boundary detection in conversational

speech. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 545–548, 2006.

S. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

J. Rocchio et al. Relevance feedback in information retrieval. *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323, 1971.

G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report 97-881, Cornell University, 1987.

A. Singhal. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24:35–43, 2001.

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

A. Singhal, C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, 1996.

A. F. Smeaton. Using NLP or NLP resources for information retrieval tasks. In T. Strzalkowski, editor, *Natural Language Information Retrieval*, pages 99–111. Kluwer Academic Publishers, Dordrecht, NL, 1999.

M. Smucker and J. Allan. An investigation of dirichlet prior smoothings performance advantage. Technical report, Technical Report IR-391, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts Amherst, 2005.

M. Smucker and J. Allan. Lightening the load of document smoothing for better language modeling retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 699–700, 2006.

M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th conference on Information and knowledge management (CIKM)*, pages 623–632, 2007.

F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings of the eighth international conference on Information and knowledge management (CIKM)*, pages 316–321, 1999.

K. Sparck Jones. Summary performance comparisons trec-2 through trec-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, 2000.

K. Sparck Jones. What is the role of NLP in text retrieval? In T. Strzalkowski, editor, *Natural language information retrieval*, pages 1–21. Kluwer Academic Publishers, Dordrecht, NL, 1997.

K. Sparck Jones. Summary performance comparisons TREC-2 through TREC-7. In *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*, 1999.

K. Sparck Jones, S. Walker, and S. Robertson. A probabilistic model of information retrieval: development and comparative experiments (parts i and ii). *Information Processing and Management*, 36:779–840, 2000.

M. Srikanth and R. Srihari. Biterm language models for document retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 425–426, 2002.

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

T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *Proceedings of the 2006 joint conference of Human Language Technologies and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 407–414, 2006.

C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

X. Yi and J. Allan. Evaluating topic models for information retrieval. In *Proceedings of the 17th conference on Information and knowledge management (CIKM)*, 2008.

H. Zaragoza, D. Hiemstra, and M. Tipping. Bayesian extension to the language model for ad hoc information retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–9, 2003.

C. Zhai. A brief review of information retrieval models. Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign, 2007.

C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th conference on Information and knowledge management (CIKM)*, pages 403–410, 2001.

C. Zhai and J. Lafferty. Two-stage language models for information retrieval. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2002.

C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.