University of California
Los Angeles

Addressing the Challenges of Underspecification in Web Search

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Michael Jason Welch

2010

The dissertation of Michael Jason Welch is approved.

Mihaela van der Schaar

Wesley W. Chu

Alfonso F. Cardenas

Junghoo Cho, Committee Chair

University of California, Los Angeles

2010

To my family and friends

Table of Contents

List of Figures

List of Tables

Acknowledgments

My advisor, Junghoo Cho, has provided incredible support and guidance throughout the course of my studies. My work has immeasurably improved from his ideas, questions, suggestions, and feedback. I owe much of what I know about approaching research and presenting results to John.

I would also like to thank the members of my committee: Alfonso Cardenas, Wesley Chu, and Mihaela van der Schaar. Their interest and expertise in my research topics fostered valuable discussions and feedback on my dissertation.

I am indebted to the many outstanding engineers and researchers whom I have worked alongside during my years as an intern with the Advanced Technology Labs at Adobe: Larry Masinter for guiding me through my first attempt at industry research, Roger Webster and Jerry Hall for insights into the world of software engineering, and Walter Chang for his guidance, feedback, and tireless efforts in seeing my work become part of shipping products; to my managers at Adobe: Ramin Behtash for allowing me to formulate my own projects, and Tom Jacobs and Tom Malloy for making my academic progress an important factor in my projects with ATL; and to the countless others whose impromptu hallway conversations and meetings influenced my work over the last six years.

Lastly, I would like to thank my many friends and classmates from the lab: Uri Schonfeld, Chu-Cheng Hsieh, Albert Lee, Barzan Mozafari, Richard Sia, Susan Chebotariov, Carlo Curino, Felix Gao, Amruta Joshi, Sung Jin Kim, Nikolay Laptev, Hamid Mousavi, Chuong Nguyen, and Alex Shkapsky. The countless discussions about research problems helped guide and improve my work. The conversations and time spent with them away from the lab made the long road to writing this dissertation immensely more enjoyable.

Vita

1981	Born, California
2000–2003	Software Engineering Intern, Caminosoft Corporation, Westlake Village, California
2003	B.S. in Computer Science and Engineering, University of California, Los Angeles
2003–2004	Graduate Student Researcher, Computer Science Department, University of California, Los Angeles
2004–2008	Teaching Assistant, Computer Science Department, University of California, Los Angeles
2004–2010	Research Intern, Systems Technology Lab, Advanced Technology Labs, Adobe Systems Incorporated, San Jose, California
2005	M.S. in Computer Science, University of California, Los Angeles
2009	Graduate Student Researcher, Computer Science Department, University of California, Los Angeles

Publications

Larry Masinter and Michael Welch. A System for Long-Term Document Preservation. *In Proceedings of IS&T Archiving 2006, May 2006.*

Michael J. Welch and Junghoo Cho. Automatically Identifying Localizable Queries. *In SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 2008.*

Michael Welch, Junghoo Cho, and Walter Chang. Generating Advertising Keywords from Video Content. *In CIKM 2010: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, October 2010.*

Abstract of the Dissertation

Addressing the Challenges of Underspecification in Web Search

by

Michael Jason Welch

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2010

Professor Junghoo Cho, Chair

The World Wide Web contains information on a scale far beyond the capacity of manual organization methods. Web search engines help users sift through that information to find data of interest through keyword searches, while also driving a multi-billion dollar advertising industry on the Web. Searching through all of the data on the Web to find the most relevant content is an enormous task, often exacerbated by underspecification and ambiguity in the queries posed by users or the underlying data itself. Users frequently omit relevant context or submit multifaceted queries, authors rarely provide explicit keywords or categorizations, and content is often missing relevant keywords. Uncertainty leads to inherent difficulty for search engines to find the best information for a particular user and query.

We investigate these problems and propose techniques to effectively satisfy the needs of users and advertisers when a search engine encounters such uncertainty. The main challenges we address consist of:

(1) Discovering which queries or keywords may benefit from contextualization.

We propose a framework for automatically identifying geo-localizable queries, establishing several features measurable from search query logs which enable traditional machine learning algorithms to classify queries with high accuracy.

(2) Given an ambiguous query, determining the most likely user requirements for each of the possible subtopics and then selecting a diverse set of pages to satisfy the greatest number of users. We describe a model for user satisfaction with a returned set of pages and propose a greedy algorithm for diversifying search results tailored towards the requirements of informational queries, when users frequently require more than one relevant result. We demonstrate notable improvement over current ranking strategies.

(3) Identifying the pertinent keywords from sparse or imprecise content. We study two approaches for generating keywords from the text content of videos and investigate related term mining approaches to overcome potential mismatches between these keywords and the keywords chosen by searchers or advertisers. We perform extensive evaluations to highlight under what conditions each method generates the most relevant keywords.

This dissertation presents and evaluates methods and algorithms which may benefit search engines, their users, and their advertising partners for a significant fraction of search instances and exabytes of data.

CHAPTER 1

Introduction

For years, written text such as books and periodicals were the primary source of information for people throughout the world. As the amount of available text increased, it became clear that organizational systems were necessary to help users find the information they seek. Classification systems such as the Dewey Decimal system organize information into a hierarchy of categories, allowing the user to browse a well-defined ontology to find sources relevant to their need.

The World Wide Web, however, has rapidly assumed the role of primary information source for people throughout the world. The enormous scale of data available on the Web renders the manual organizational techniques from the printed world infeasible. The paradigm for finding information has thus shifted from primarily organized, faceted lookups to keyword search. While this potentially simplifies the task for users, it introduces numerous challenges for the search systems designed to retrieve and present information based on these keywords.

1.1 Challenges of Underspecification

Web search engines work towards a key objective: returning the most relevant content for each query. Many difficulties facing Web search engines stem from underspecified queries and data. When a query can be interpreted multiple ways, precisely determining the user's true intent is a difficult problem. Likewise, iden-

tifying the correct context and keywords from textual and multimedia content on the Web can be challenging, particularly when that content is short or incomplete. Manual indexing or classification is impractical given the vast scale of data on the Web, authors rarely provide explicit categorizations for their content, and users often fail to fully articulate their needs, placing the onus on the search system to solve these problems automatically. These issues lead to an inherent difficulty for search systems to find the right information for a particular user and query.

Although queries are supplied as keywords, data on the Web is not limited to text. The Web is quickly evolving as a medium for distribution of both professionally produced and amateur, user generated videos. Multimedia content on the Web has rapidly gained popularity in the past few years, with recent data from the Web analytics company Alexa [Ale] indicating video sharing site YouTube [You] as the third most visited site globally. Viewers in the United States alone watched an estimated 25.3 billion videos totaling 1.56 billion hours during the month of August 2009 [com]. Research estimates suggest nearly one billion users will watch online videos by 2013 [Res]. Along with this surge in viewers and available content comes the need for effective, automatic methods for identifying relevant keywords for the video data.

A Web search engine incurs significant costs maintaining the necessary resources to operate and serve users. They typically generate revenue to finance their operations by displaying advertisements, both alongside content and with search results. Identifying the relevant keywords for content leads to a better search experience for users, and improving user satisfaction can increase traffic, which is important for optimizing advertising revenue. At the same time, the relevance of the content to the advertisement has a significant impact on the

effectiveness and therefore profit generated from the ads, and advertisers are typically willing to pay more when pertinent context, intent, and demographics are known about a user viewing content or issuing a search query.

Search engines can provide better results for users and increase their profits by addressing the same challenges associated with underspecification and uncertainty. We illustrate some of these challenges further with a few examples:

- **Context Awareness**. Consider the task of finding information about water conservation laws for a person's home city. Prior to the Web, a user would likely contact their local water department or city hall directly and ask for pamphlets or announcements regarding current regulations. On the Web, a user would likely issue the query "water conservation laws" to a search engine and expect to find the same information. The Web scenario, however, is at a significant disadvantage. Critical pieces of information implicitly available in the traditional case have not been provided by the user. In particular, the search engine does not understand the semantics of "water conservation laws", or that such information is relevant on a city by city basis. If the search engine were able to determine that the user's current city is important context for the query, it may use that information to serve more relevant pages and advertisements.

- **Search Diversification**. A user is interested in obtaining information about symptoms related to viral infections. To find relevant information offline, a user may browse a classification system to find books or periodicals related to health, science, and medicine. Online, a user may search with a reasonable query, such as "virus symptoms" and have difficulty finding relevant information due to the ambiguity of the term "virus". Virus has many senses, including biological viruses, computer viruses, several film

titles, and so on. In the offline scenario, the user avoided any ambiguity by navigating an explicit topical hierachy. Current Web search engines have difficulty automatically resolving ambiguity in queries or Web pages, and search results are often dominated by a single interpretation of the ambiguous term, leaving users whose intent was another subtopic unsatisfied. If the search engine can estimate the user's intent and information need for each subtopic, it may be able to present results from multiple subtopics and satisfy more users.

- **Identifying Relevant Keywords**. A television network intends to publish episodes from their catalog on the Web. Facilitating discovery and driving traffic to the content through online search requires identifying a set of highly relevant keywords for the videos. Given the quantity and duration of the videos, manually specifying these keywords is impractical. Even with an available text source for a video, such as a closed captioning track or speech transcript, errors or omissions in the text and the lack of structure hinder the application of current methods for relevant keyword identification from Web pages [YGC06], which often rely on explicit markup languages and formatting style. Furthermore, the *vocabulary impedance problem* [RCG05] between keywords from the video content and searcher or advertiser supplied keywords makes it more difficult for users to find their desired content and results in missed opportunities to place relevant ads. If the search engine can make better use of the available text to generate a larger and more accurate set of keywords for the content, it can build a more comprehensive index to improve search relevance and advertising coverage.

1.2 Outline of Dissertation

In this dissertation, we address problems associated with underspecification which lead to the Web search challenges highlighted in the above examples. From the query perspective, we investigate how the search engine can determine when search keywords imply additional context and what a search engine can do for ambiguous queries when user intent is unknown. From the content perspective, we study how the search engine can identify relevant keywords when the only text sources available are plaintext, short, and possibly contain errors or omissions. The chapters of this dissertation are organized as follows.

1. **Chapter 2: Identifying Implicit Query Context.** Users often submit queries to a search engine with a clear informational need, but may omit an implicit context necessary to fully capture their intent. We study how a search engine can determine which queries carry such an implicit intent. Studies suggest that 30% of queries issued to a Web search engine are geo-localizable, or dependent on a user's geographic location. However, users only explicitly add localizing keywords to such queries about one half of the time [WC08].

   In this chapter we present and evaluate a framework for identifying geo-localizable search queries based on large scale query log analysis and machine learning techniques. We describe key properties of localizable queries and establish several features computable from Web search logs which help differentiate between localizable and non-localizable queries. We then evaluate a variety of popular supervised classification algorithms using the computed feature scores. We also show that significant gains in classification accuracy can be achieved by combining multiple classifiers in a majority

voting scheme.

2. **Chapter 3: Ambiguous Queries.** Ambiguous queries present a serious challenge to Web search engines and, like localizable queries, represent a significant fraction of query instances [San08]. With current approaches the top results for these queries tend to be homogeneous, making it difficult for users interested in less popular aspects to find relevant documents. We investigate the problems search engines face when dealing with uncertainty in user intent, identifying three important factors controlling how a search engine may best allocate its result slots to pages such that we satisfy the maximum number of users under such conditions:

   - User intent: when presented with an ambiguous query, such as "virus", we study how the search engine can probabilistically determine which subtopic(s) an average user is most likely to be interested in.
   - Page categorization: given a page which matches the ambiguous query term(s), we investigate how the search engine can approximate the likelihood the page satisfies a particular subtopic.
   - Pages required: we examine how many relevant pages a user is expected to visit, and how that information should influence search result allocation strategies.

   In Chapter 3, we study result diversification as a strategy for supporting ambiguous queries. We present a model for user satisfaction with a set of search results which explicitly considers that a user may need more than one page to satisfy their need, making it particularly suitable for *informational* queries. This modeling enables our proposed search diversification algorithm to make a well-informed tradeoff between a user's desire for *mul-*

tiple relevant documents, probabilistic information about an average user's interest in the subtopics of a multifaceted query, and uncertainty in classifying documents into those subtopics. We evaluate the effectiveness of our algorithm against commercial search engine results and other modern ranking strategies, demonstrating notable improvement in multiple document scenarios.

3. **Chapter 4: Keyword Generation.** With the proliferation of online distribution methods for videos, search engines and video owners require easier and more effective methods for identifying relevant keywords for video content, both to improve content discovery through keyword-based video search and for contextual advertising. While many pages on the Web contain internal and external clues for determining meaningful keywords for the page, such as HTML markup and anchor text from neighboring pages, identifying keywords from video content remains a difficult task. Current methods for selecting relevant keywords for videos are limited by reliance on manually supplied metadata.

   In this chapter we study keyword generation for videos from accompanying text sources, such as scripts, closed captioning tracks, and speech transcripts. We address several challenges associated with using such data. To overcome the high error rates prevalent in automatic speech recognition and the lack of an explicit structure to provide hints about which keywords are most relevant, we use statistical and generative methods to identify dominant terms in the source text. To overcome the sparsity of the data and resulting vocabulary mismatches between the source text and search queries or keywords selected by an advertiser, these terms are then expanded into a set of related keywords using related term mining methods,

enabling the search engine to retrieve content for search queries and match advertiser specified keywords which do not directly appear in the original data [WCC10].

We present a comprehensive analysis of the relative performance for statistical and generative methods across a range of text sources and videos, including professionally produced films, news clips, and amateur or user generated videos. Our evaluations consider both the relevance of the keywords to the content, and the usefulness of those keywords for advertising.

CHAPTER 2

Identifying Implicit Query Context

2.1 Introduction

Typical queries submitted to Web search engines contain very short keyword phrases [JSS00]. These short text queries are often insufficient to specify a user's complete information need, yet users still have an expectation of finding relevant content. With Web search engines handling billions of queries per month, and considerable financial incentives to increase market share, maximizing user satisfaction with query results is in constant focus.

Personalizing search results to an individual by incorporating contextual metadata about the user during document retrieval and result ranking are well studied approaches to improving overall user satisfaction. Before personalizing results for a particular query, however, the search engine must identify what context is relevant. In this chapter we address the following question: How can the search engine know when metadata about the user is applicable to a particular *query*? When making the decision to automatically contextualize a query, it is important to avoid incorrectly applying irrelevant context. In particular, we wish to avoid "false-positives", which would lead to erroneously contextualizing queries.

In this chapter we present a technique for automatically identifying a class of queries we define as *localizable* from a Web search engine query log. *Localizable* queries are those search queries for which the user would implicitly prefer to see

results prioritized by their geographical proximity; "airport shuttle" or "Italian restaurant", for example, are queries likely submitted by a user with the goal of finding information or services relevant to their current city.

Our analysis suggests a significant fraction of user queries would benefit from such localization. Users, however, explicitly add location constraints to less than one half of such queries. By determining when it should automatically localize a particular query, the search engine can not only improve the user's search experience, but also generate additional revenue by enabling advertisers and businesses to more effectively target their local audience.

We perform analysis on a large scale query log, identifying the queries which contain locations as contextual modifiers and extracting the base portion of those queries. We describe a set of distinguishing features and experiment with a variety of classifiers and supervised learning algorithms to automatically identify the set of localizable queries. Cross validation experiments are used to evaluate the effectiveness of these features and classifiers.

2.2 Motivational Studies

Before we discuss the technical aspects of our work, it is worth spending a moment to focus on a few important preliminary questions. In particular, we wish to verify that the concept of "localizable" is *consistent* amongst users, and that automatic localization is *worthwhile*.

2.2.1 Query Coverage

Web search engines typically incorporate a complex mix of factors when ranking query results. Factoring in localization increases the complexity of the system,

and so it is important to verify that its overall impact justifies that complexity. To that end, we conducted a user study to estimate the percentage of query instances which would benefit from localization. We gave 9 survey participants each a different list of 100 randomly sampled entries from a search query log and asked them to classify each query into one of three categories:

1. Query would likely not benefit from localization
2. Query would likely benefit from automatic localization
3. Query is already localized

Our survey participants said that, on average, 70% of queries would not benefit from localization, 16% of queries would benefit from automatic localization, and 14% of queries were already localized. Automatic localization may potentially improve the results for a significant fraction of user queries, as these results suggest that, while approximately 30% of the queries issued to search engines are localizable, users only explicitly localize about one-half of them.

2.2.2 User Agreement

Research involving user satisfaction and search result personalization typically must deal with some level of subjectivity. Automatic localization is a form of personalization, and so we pose the question: *do users generally agree on which queries should be localized*? To address this issue, we administered a second user survey. As our goal is to now see whether users agree on which queries are localizable, and approximately 15% of queries in the log are localizable, we felt a random sample from the query log would not provide sufficient opportunity for users to disagree. We constructed a list of 102 queries, approximately one half of

which we believed to be localizable. This list was presented to 8 participants who were asked to make a binary judgment for each query about whether it would benefit from localization or not.

The results were tabulated to determine whether users agree on which queries would benefit from localization. Figure 2.1 shows a plot of user agreement, with the number of users who *disagreed* with the majority along the X-axis, and the number of queries on the Y-axis. We found that users agreed that queries for goods and services, such as "food supplies" and "home health care providers" were localizable, while more general queries for information, such as "calories in coffee" and "eye chart" are not. The intent of other queries, such as "medical license" and "marathon" are more vague, and our survey participants were evenly divided.

We note that participants were asked to make their best interpretation of the query intent given only the query text, and so some level of discrepancy is expected. The overall results are encouraging, as we see that users are evenly divided on only 8 of the queries, while at most one person disagreed with the majority for about 50% of the queries.

2.2.3 Preliminary Results

Without a complete implementation of a "localizing" search system to perform experiments with, we must find other ways to estimate the overall user satisfaction with query localization. User studies by Joachims et. al. [JGP05] suggest that clickthroughs are a reasonable approximation of relevance feedback. During query feature collection, we measured the clickthrough rates for both the localized and non-localized form of the same "base" query, and found that the average clickthrough rate for the localized instances of a localizable query is ap-

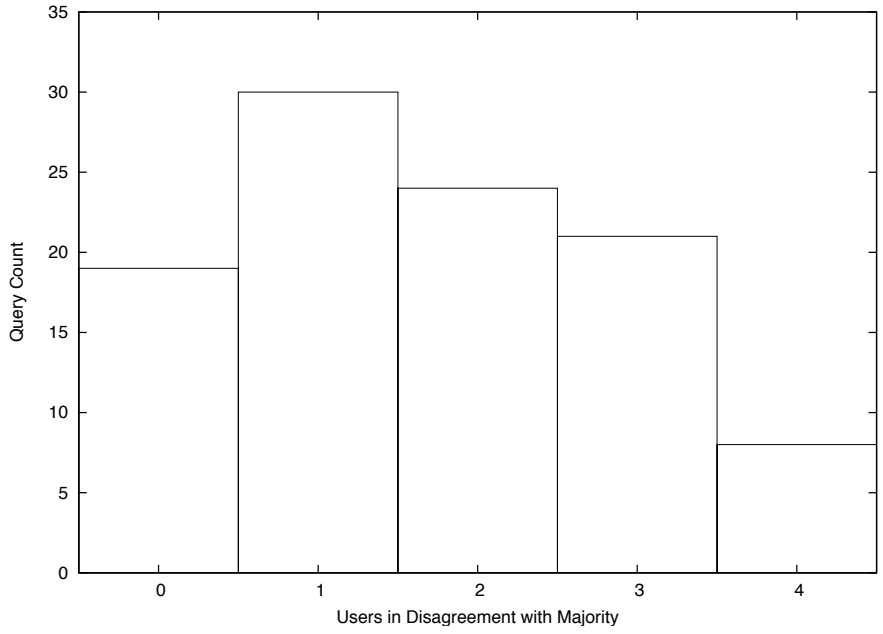

Figure 2.1: User agreement survey

proximately 17% higher than for the non-localized instances. While not a perfect interpretation of user preference, it is a promising indicator.

2.3 Overview

For any query q, we wish to efficiently determine whether q is localizable or not. Our basic approach is to build a query classifier using features collected from a Web search engine query log. This classifier can then be used by the search engine to make realtime decisions about whether location is meaningful context on a per-query basis.

We begin identifying localizable queries by finding previously issued queries which contain an explicit localization modifier, with the assumption that the "base" of these queries may be generally localizable. We identify all entries in

the query log which contain "locations" and extract the base of these queries. Once we have the set of all base queries, we select a sample to use for classifier training. For this subset of queries we compute relevant distinguishing features and evaluate multiple well-known supervised classifiers to determine which are best suited for our task. Each of these steps are discussed in the following sections.

2.4 Query Log Analysis

In our analysis we use a search query log from America Online [PCT06], which contains queries from 657,426 distinct users over a three month period from March 1 to May 31, 2006. The log contains approximately 36 million rows of data, covering 10 million textually unique queries from 21 million search instances.

We start construction of a classifier by finding queries in the log which contain location modifiers. The query log contains queries in the English language, and so we have focused our location identification on states, counties, and cities in the United States using a list available from the U.S. Census Bureau [Bur]. For queries which contain one or more of these locations, we consider the location as a contextual modifier added by the user, and remove it to find the "base" query. For example, the base of the query "san francisco public parks" is "public parks". These base queries are the ones which we would like to automatically localize.

In the remainder of this section, we will discuss how base queries are identified from the query log, how queries sharing a "similar" base are grouped together, and some considerations for matching user queries to entries in the log.

2.4.1 Identifying Base Queries

Queries are typically very short, consisting of only 2-3 terms [JSS00] rather than complete, grammatically correct sentences. Additionally, all queries in the log have been normalized to lower case. As a result it is difficult to employ natural language processing (NLP) approaches, such as parts-of-speech tagging, to aid location tagging. Likewise, techniques based on other indicators, such as capitalization or punctuation, may not reliably label locations within queries.

Instead of using cumbersome grammatical tools to identify locations, we utilize a simple string matching process, and ensure accuracy using a set of features carefully selected to eliminate false positives. While relatively straightforward, this technique proves quite effective, and the simplicity supports scalability as well as language independence, both of which are requirements for Web search engines.

To identify localized queries, we inspect the text of each query and compare it to the Census Bureau list of locations. Every match generates a new base query, where the matched portion of text is tagged with the detected location type (state, county, or city). Queries may contain multiple localizations, such as a city and state name. Rather than complicate the tagging process, we choose to simply remove each tagged token and enqueue the remainder of the query for further processing. As a result, a single entry in the query log may produce multiple base queries.

We favored this technique over removing all "locations" and generating a single base query from each entry because, in general, we cannot be certain when query terms are specifying a location. Several words in the English language are also used as location names, such as the city of Parks, Arizona. If we choose to remove all terms matching a location in a single step, we would not be able

Query ID	Base Query	Location Tag
10005397	county florida animal shelter	city:lee
10005397	——— county animal shelter	city:florida
10005397	——— county animal shelter	state:florida
10005397	florida animal shelter	county:lee county
10005397	——— animal shelter	city:florida
10005397	——— animal shelter	state:florida
10005397	lee county animal shelter	city:florida
10005397	——— county animal shelter	city:lee
10005397	——— animal shelter	county:lee county
10005397	lee county animal shelter	state:florida

Table 2.1: Base query generation

to identify the correct base "public parks" in the example "san francisco public parks" discussed above.

Table 2.1 shows all of the base queries generated from the source query "lee county florida animal shelter". Indentation is used to illustrate how the original query is processed to ultimately result in each of the possible base queries shown. For example, the first row is obtained by removing "lee" from the original query. The second and third rows are generated by further tagging the resulting base query "county florida animal shelter". The distinct base queries and their associated location tags generated by this processing are summarized in Table 2.2. For the approximately 10 million distinct entries in the query log, we identify 4.9 million unique base queries.

Base Query	Location Tag(s)
county florida animal shelter	city:lee
county animal shelter	city:florida, city:lee, state:florida
florida animal shelter	county:lee county
animal shelter	city:florida, county:lee county, state:florida
lee county animal shelter	city:florida, state:florida

Table 2.2: Base query tags

2.4.2 Base Query Grouping

As Table 2.1 shows, tagging the query "lee county florida animal shelter" generates 10 entries comprising 5 textually distinct base queries. After processing the entire query log, we group together queries which share a "similar" base query q_b, and define $L(q_b)$ as the set of location tags which occur with q_b. We explored several alternatives for this similarity mapping, ranging from an exact string match to a bag-of-words model with stopwords eliminated and terms stemmed using Porter's suffix stemming algorithm [Por97].

The choice of mapping function has implications on the accuracy and coverage of our classifier, as well as how we determine which base query a potentially localizable user query issued to a search engine corresponds to. We will now briefly discuss some of the options considered.

2.4.2.1 Exact Match

An exact match model produces the largest set of distinct base queries, as we only group together the entries that are textually equivalent. Using the entire text of a query allows us to distinguish between semantically different queries whose text may, from an algorithmic point of view, only differ in seemingly insignificant

ways.

Exact matching, however, also potentially introduces many unintelligible base queries. With location terms removed from a query, the remaining text may never actually be issued to a search engine as a query by itself. Extracting the base of the query "parks in [city]" would produce "parks in", which is unlikely to appear as a user query, and in fact does not appear in the query log. The query "parks", however, occured 53 times.

2.4.2.2 Stopword Elimination

Many modern information retrieval systems ignore the most common words, such as conjunctions and prepositions, frequently referred to as stopwords. By eliminating stopwords from queries, we more easily group together logically equivalent queries, such as "parks in [city]" and "parks near [city]" into a single base "parks".

2.4.2.3 Bag of Words

A bag of words model ignores the ordering of terms in a query. Combined with stopword elimination, this may help consolidate semantically equivalent queries such as "airport shuttle" and "shuttle to airport" into a single common base. In some cases, however, ignoring the word ordering may actually change the meaning of the query.

2.4.2.4 Term Stemming

Term stemming algorithms, such as the suffix stemming algorithm described by Porter [Por97], can help normalize term tense and plurality. While this may improve precision for some semantically equivalent queries such as "restaurant"

Stopwords Eliminated	Bag-of-words	Stemmed	Distinct Queries
No	No	No	4,898,589
Yes	No	No	3,940,233
Yes	Yes	No	3,808,215
Yes	No	Yes	3,790,692
Yes	Yes	Yes	3,640,513

Table 2.3: Base query grouping

and "restaurants", it may also occasionally result in collisions between distinct terms which share a common stem, reducing precision. For example, "universe" and "university" share the common stem "univers".

Other forms of stemming, such as morphological analysis and lemmatization, may produce more accurate results for related term grouping than algorithmic suffix stemming. Lemmatization is frequently discussed in the field of statistical machine translation [Lee04]. Such techniques are significantly more complex, however, typically requiring additional data sources such as a lexicon and part-of-speech (POS) tagger.

2.4.3 Evaluation

In our classifier evaluations, we find that stopword elimination is the only pre-processing step which has significant impact on the final classification results. Fundamentally, it provides the best combination of normalizing logically equivalent queries with minimal semantic loss. As Table 2.3 shows, additional processing would not noticeably reduce the size of the base query set, and as a result, calculated feature scores will not change significantly. In the remainder of this chapter, when we refer to the base of a localized query, we are referring to the

stopword eliminated version.

2.5 Feature Selection

The tagging process generates a base query any time it finds text which matches a location. Several city and state names are homographs, and thus text matching is not sufficient. For example, "kansas" may refer to the state, one of several cities, the rock band, or even a movie with that title. When we find a query containing "kansas", how can we know whether the user was referring to a location or one of the other senses of the word?

In this section we discuss a set of features measurable from a query log which a supervised classifier may use to make that determination, and discriminate localizable queries from the false positives. We investigated several features, both about the individual queries as well as aggregate measures of the grouped queries. Some query features, such as frequency counts, have relatively straightforward interpretations. Others are more subtle and require additional discussion.

Although the analysis here is performed over a window of user queries contained within a log, query logs collected by search engines are constantly expanding. This necessitates the ability to adapt classifications as new examples are collected, and it is important to consider this when selecting features. We therefore focus feature selection on those which are easily recalculated as the data expands.

2.5.1 Localization Ratio

Online information sources such as Wikipedia [Wik] rely on the collective expertise of their users to ensure the knowledge base is accurate. We adapt a similar

model for Web search queries, where every query instance can be treated as a "vote". In our case, users vote for the localization of query q by submitting it to the search engine with a location specified.

For every textually distinct query q_i we define $b_i \in [0, 1]$ as the fraction of users who would benefit from the localization of q_i. Every query q_i has an associated value $r_i \in [0, 1]$, defined as:

$$r_i = \frac{Q_i(L)}{Q_i + Q_i(L)} \tag{2.1}$$

Q_i is the count of all instances of q_i in the query log, and $Q_i(L)$ is the count of all query instances tagged with some location $\ell \in L$, for which q_i is the base. This r_i value represents the "localization ratio" for q_i. Assuming some fraction of the users who issued q_i to the search engine implicitly wanted localized results, r_i defines an estimated lower bound on b_i.

Localization ratio provides some insight into what fraction of users believe that a particular query would benefit from localization. It is, however, susceptible to small sample sizes, as a query issued by a single user may have an r_i value of 1. Localization ratio is also unable to identify false positives resulting from incorrectly tagged locations. For example, the base query "barnes" has an r value of over 0.75 for the data in the log, yet a vast majority of its occurrences come from incorrectly tagging "noble" as a location in the query "barnes and noble". For these reasons, localization ratio is insufficient as the sole feature for classification.

2.5.2 Location Distribution

The query tagging process is based on string comparison, creating a match for any text which is listed as a US state, county, or city. Some of these locations are

Location Tag	Count
city:independence	156
city:homestead	5
state:texas	4
city:lincoln	2

Table 2.4: Locations occurring with "declaration"

homographs in the English language, introducing false positives to the candidate list which must be filtered out. As an example, "Independence" is a city in Missouri, and is tagged as such in the query "declaration of independence". The base query "declaration" occurs a total of 176 times with some location, 113 alone of which are due to the query "declaration of independence". Table 2.4 shows the top 4 most frequently occurring locations for the base query "declaration".

To aid in identifying these entries, we start with a basic assumption about any localized query q_l:

$$\forall \ell \in L(q_b) \Pr[\ell \in q_l | q_b = base(q_l)] \approx \frac{1}{|L(q_b)|} \tag{2.2}$$

That is, given an instance of any localized query q_l with base q_b, the probability of q_l containing location ℓ is approximately equal across all possible locations of q_b. Base queries which have a highly skewed distribution of location occurrence counts suggest that either the query is only relevant to those locations, or the tagged "location" is actually part of the query, rather than a localization modifier.

To estimate the distribution, we calculate several measures for the set of locations $\ell \in L(q_b)$, including minimum, maximum, mean, median, and standard deviation of their occurrence counts.

2.5.3 Clickthrough Rates

Search result clickthrough rates have been used in studies evaluating and improving the effectiveness of Web search engines [Joa02]. User studies suggest that clickthroughs are a reasonable approximation of relevance feedback [JGP05]. These studies focus on improving the document ordering by treating user clickthroughs as relative relevance judgments and adjusting for any bias the presented ordering introduces.

We are primarily concerned, however, with the relevance of the overall result set, as opposed to the relative ranking of the returned documents. In our experiments, we use clickthrough events recorded in the query log as a comparator of user satisfaction with the localized and non-localized versions of their search results. We define a binary user satisfaction function with the results of query instance q_i as:

$$S(q_i) = \begin{cases} 1 & \text{if at least one result was clicked on,} \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

We compute the total clickthrough count for each base query q_b as the sum of S over all instances of q_b in the query log. This sum is then divided by the number of instances of q_b to calculate the clickthrough rate. Likewise, we calculate the clickthrough count and rate for the set of localized queries with base q_b. We may then compare the clickthrough rates of both the localized and base forms of a query, where a positive difference is considered as an indicator of increased user satisfaction with the results.

2.5.4 Frequency Counts

At first glance, frequency count measures may seem like potential red herrings, given that there is no bound on their values. For example, the query "barnes and noble" is tagged as "barnes and [city:noble]" and "[city:barnes] and noble", due to the cities of Noble, Oklahoma and Barnes, Kansas. The query "barnes and noble" occurs in 2679 unique search sessions, significantly higher than the average of just over 2 occurrences for a unique query. If we judged localizabilty based on a threshold of popularity, both of the base queries "barnes" and "noble" would likely be incorrectly identified as localizable.

Despite this, frequency counts are still useful measures. In particular, frequency count serves as a normalizing or significance factor for other features, such as r, by taking into account a query's popularity.

2.5.4.1 User Distribution

In addition to query occurrence counts, we also consider the sources for those occurrences. For every query q_i, we calculate the number of distinct users who have issued the query, in both its localized and non-localized forms. These two measures provide a different form of normalization for the occurrence counts q and q_L by adjusting for bias introduced from a single user issuing the same query multiple times.

2.6 Experimental Results

Using the features discussed in the previous section, we now evaluate classification algorithms for learning localizable queries. We manually tagged a training data set and evaluated the effectiveness of several supervised learning algorithms,

Symbol	Definition
q	Occurrence count of a query
q_L	Localization count of a query
r	Localization ratio
n_L	Number of distinct locations occurring with q
$\bar{n_L}$	Average occurrence count for the n_L locations of q
$\tilde{n_L}$	Median occurrence count for the n_L locations of q
σ_{n_L}	Standard deviation of occurrence count for the n_L locations of q
$n_{L_{min}}$	Minimum occurrence count for the n_L locations of q
$n_{L_{max}}$	Maximum occurrence count for the n_L locations of q
u_q	Number of users who issued the query q
u_{q_L}	Number of users who issued the localized query q_L
c_q	Number of clickthroughs for query q
c_{q_L}	Number of clickthroughs for localized query q_L
$\bar{c_q}$	Normalized clickthrough rate for query q
$\bar{c_{q_L}}$	Normalized clickthrough rate for localized query q_L

Table 2.5: Summary of features

including naive Bayesian classifiers, decision trees, support vector machines, and neural networks. In addition to these individual classifiers, we evaluated techniques for improving accuracy by combining multiple classifiers, including AdaBoost [FS96], Bayesian boosting [KHZ00], and independent majority voting.

Our experiments were conducted using classifiers and boosting techniques implemented as part of the RapidMiner [MWK06] machine learning framework.

2.6.1 Training Data

In order to eliminate the most impertinent data, we performed two filtering steps when selecting our training data set. Prior to selection, we removed candidate queries with localization count $q_L <= 1$, which reduced the size of the stopword-eliminated candidate localizable query set to approximately 1.7 million. We selected a random sample of 200 entries from this set and, prior to tagging it, performed an additional filtering step as follows: we removed queries which occurred with only one distinct location modifier ($n_L <= 1$), were only issued by a single user ($u_q = 1$), or whose base form was never issued to the search engine ($q = 0$).

After this filtering, we manually tagged the 102 remaining entries from our random sample of candidate localizable queries, 48 of which were deemed to be localizable. The training set consisting of 48 positive (localizable) and 54 negative (non-localizable) examples was used in a series of classification experiments discussed below. This set comprises the same queries presented to users in our survey discussed in Section 2.2.2, where our classification agreed with the majority for 91 of the 102 entries.

2.6.2 Classifier Evaluation

We compare the effectiveness of several well-known supervised classifiers using standard precision and recall measures. As our overall goal is to identify localizable queries and ultimately use that knowledge to automatically apply user context when retrieving and ranking search query results, we feel it is important to emphasize precision over recall, and in particular, the precision of positive (localizable) classifications. In terms of user satisfaction, correctly localizing a smaller subset of all localizable queries is preferable to localizing a larger subset at the expense of increasing the number of incorrectly localized queries.

The precision and recall measures discussed below are for positive example identification based on 10-fold cross-validation experiments. To compensate for our filtering step on the training data, we consider the queries removed to be classified as non-localizable. While this filtering does not affect the computed precision, based on our survey in Section 2.2.1 we approximate 15 of these 98 queries are localizable, and adjust the recall score accordingly.

2.6.2.1 Naive Bayes

Using a set of (assumed independent) feature scores, a naive Bayesian classifier estimates the probability a given instance belongs to each of the possible discrete output classes. In our case, each instance is a user query, and the output is a boolean variable specifying whether the query is localizable or not. Despite simplistic independence assumptions, naive Bayes classifiers typically perform comparably to more complex classifiers [Ris01]. For our data set, the naive Bayes classifier achieves 55% precision at 59% recall.

In addition to feature independence, naive Bayes classifiers assume contin-

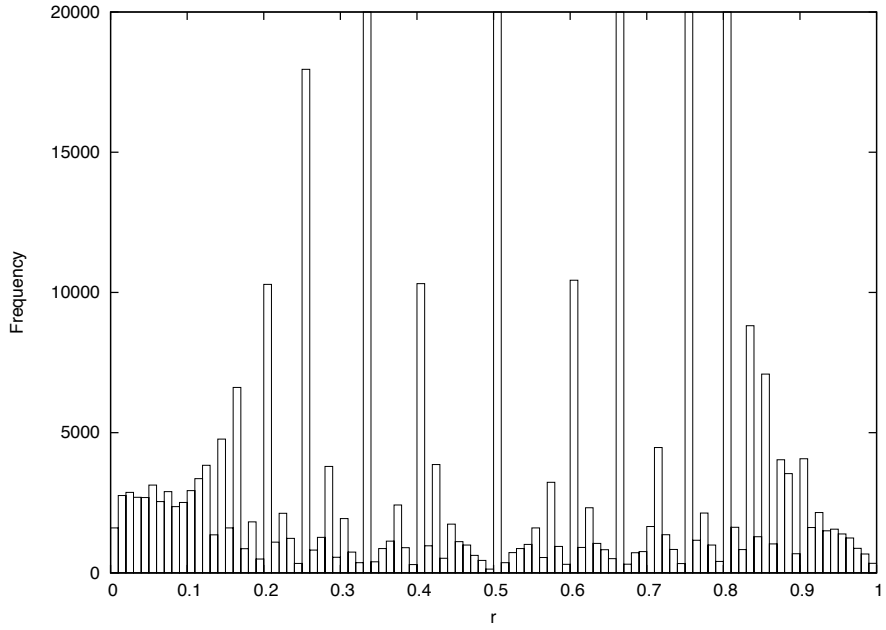

Figure 2.2: Localization ratio (r) distribution

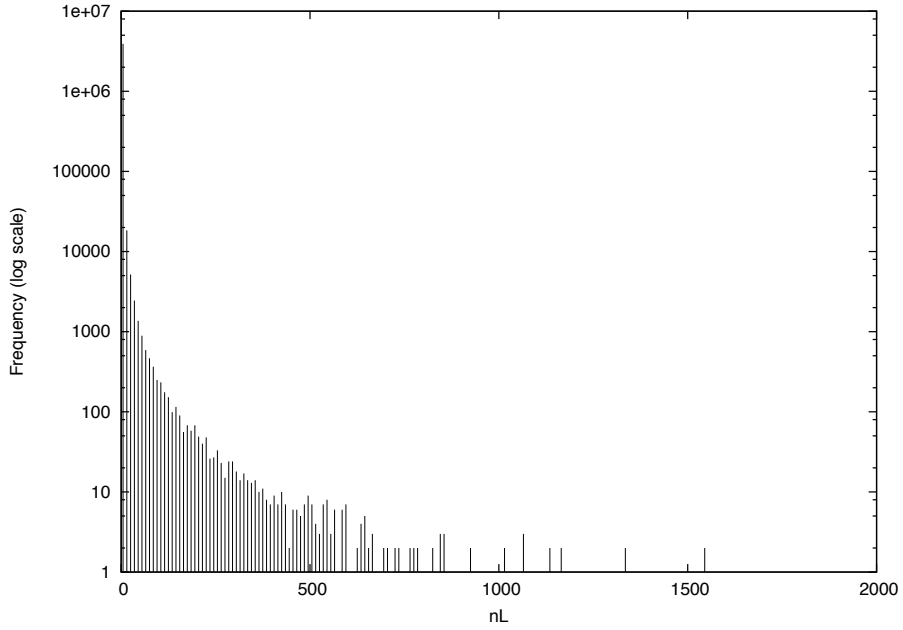

Figure 2.3: n_L distribution

Criteria	Precision	Recall
Information Gain	67%	57%
Information Gain Ratio	64%	56%
Gini Coefficient	68%	51%

Table 2.6: Decision tree performance

uous variables follow a Gaussian probability distribution. The distribution for some features, such as localization ratio (r), follows such a distribution. Other features, such as location frequency count (n_L) do not follow a Gaussian, as seen in Figures 2.2 and 2.3.

As the Gaussian assumption does not hold for all features, we investigate an alternative. Flexible Naive Bayes classifiers use a kernel-based density estimation function for continuous variables, and have been shown to greatly reduce the error rate of naive Bayes classifiers [JL95]. A kernel-based naive Bayes classifier improves the classification accuracy to 64% precision, albeit at a reduction in recall to 43%.

2.6.2.2 Decision Trees

Decision trees are widely used in data mining and machine learning applications. When constructing a decision tree, the training example set is recursively divided into subgroups based on a particular feature. In our evaluations, we construct decision trees with three distinct split criteria: information gain, the Gini coefficient, and the normalized information gain ratio. Table 2.6 shows the precision and recall measurements for each of these criteria.

A significant advantage of decision trees is the transparency of the final classifier. We inspected each of the three separate decision trees generated to study

which features were the most distinguishing. The localization ratio r was used in all three trees, as were some combination of location distribution measures (n_L, $\bar{n_L}$, and $\tilde{n_L}$). Clickthrough rates ($\bar{c_q}$ and $\bar{c_{q_L}}$) were factors in two of the three trees.

2.6.2.3 SVM

Support vector machines (SVMs) [Bur98] are a popular form of supervised learning. SVM is well suited to binary classification problems, where each instance can be represented by a set of n distinct numeric values. These n values are treated as a vector describing a point in an n-dimensional space. SVMs separate the example instances into two groups by finding the $(n-1)$-dimensional hyperplane such that the "distance" between the two groups is maximized.

Like many vector-based techniques, SVM classifiers are relatively opaque, making it more difficult to manually inspect and determine which features contributed most significantly to the classification. Regardless, the accuracy and recall of SVM for our classification task surpasses decision trees, achieving 75% precision at an 62% recall rate.

2.6.2.4 Neural Network

Neural networks are relatively complex systems capable of, among other tasks, supervised learning for classification [Bis95]. The nodes in a neural network can be separated into input and output layers, and some number of internal "hidden" layers. We evaluated feedforward neural networks comprising of one to three hidden layers, beyond which recall for positive training examples dropped to zero. Table 2.7 show the results, which indicate that neural networks are the most accurate of the individual classifiers evaluated.

Hidden Layers	Precision	Recall
1	79%	54%
2	85%	52%
3	76%	49%
4+	n/a	0%

Table 2.7: Neural network performance

Base Classifier	Precision	Recall
Naive Bayes	63%	54%
Kernel Naive Bayes	68%	44%
Information Gain	72%	57%
Information Gain Ratio	64%	43%
Gini Coefficient	67%	56%

Table 2.8: Classifiers with boosting

2.6.2.5 Boosting

Boosting algorithms, such as AdaBoost [FS96], have been shown to improve the accuracy of "weak" learning classifiers. The final "strong" classifier produced by the boosting algorithm generally consists of a weighted combination of multiple weak classifiers, iteratively trained with a weighted set of examples based on previous classification errors. We evaluated the effectiveness of BayesBoost [KHZ00] with naive Bayesian classifiers and AdaBoost with decision trees. The results were mixed, as shown in Table 2.8. In some cases, precision and recall actually decreased when applying boosting.

Decision Tree Criteria	Precision	Recall
Information Gain	94%	46%
Information Gain Ratio	90%	44%
Gini Coefficient	93%	41%

Table 2.9: Ensemble classifier results

2.6.2.6 Ensemble Classifiers

While the supervised classifiers discussed above produce relatively high precision results, we observed that the set of false positives (queries incorrectly classified as localizable) produced by the individual classifiers did not fully overlap. We experimented with another style of aggregate learner, where the final classification is determined by the majority vote from a set of discrete classifiers. Unlike boosting, which builds a final classifier from multiple instances of the same learning algorithm trained on varying example sets, this "ensemble" style classifier comprises distinct learners trained on the same example set.

We choose to combine the best individual performing classifiers using a simple majority vote scheme, where each component classifier is given equal weight. The best voting classifier achieved significantly higher precision than any individual classifier: up to 94% precision at 46% recall. Table 2.9 shows the results for three such voting classifiers, each consisting of a neural network with two hidden layers, an SVM classifier, and a decision tree with the specified split criteria.

2.6.3 Discussion

These evaluations demonstrate that conventional supervised learning algorithms are capable of distinguishing localizable queries with relatively high levels of precision. Neural networks successfully identify over one half of localizable queries

with 85% accuracy. SVMs identify a larger subset of localizable queries than neural networks, while precision decreases to 75%. Taking the majority vote of these two independent classifiers along with a decision tree, we are able to achieve over 90% classification accuracy.

2.7 Related Work

Many researchers have studied location as a context for search queries [GHL03, WWX05, JZR08, YRL09]. Some of these approaches rely on cues from external data, issuing multiple queries to the search engine [GHL03, WWX05] to extract geographic intent. Others make use of the user's IP address to identify locations or geographic intent in queries. Language modeling approaches have also been used for identifying general or specific geo-intent for queries [YRL09].

Considerable work has been done in the area of personalized and per-user Web search. Researchers have investigated several data sources for refining or expanding user queries with relevant contextual terms, including in-link anchor text [KZ04], co-occurring terms within the result set [XC96, BSA94], co-occurring terms in query logs [HCO03], and keywords from the user's desktop environment [KCM06].

Other query-dependent studies by Lau and Horvitz [LH99] use Bayesian networks to estimate user goals by classifying query refinement patterns found in search engine logs. Lee et. al [LLC05] present a set of features for automatically classifying user intent as *navigational* or *informational* for a set of queries.

Taxonomies and ontologies have been used to filter and rank search results using concept weights learned from user browsing behavior [PG99, SG05]. Liu et. al [LYM02] discuss personalizing and disambiguating queries by classifying

them into per-user category profiles based on past browsing history.

Researchers have also studied techniques for personalization focused on higher level ranking. Ideas for modifying popular Web page ranking algorithms, such as by adjusting the "jump probabilities" of the *random-surfer model* based on user preferences or trusted Web pages, were proposed in [PBM98, JW03, GGP04].

These works are primarily focused on solving the problem of *how* to contextualize results, rather than determining *when* it is appropriate to do so. Our work differs from most prior research in personalized Web search by addressing contextualization from a *query-dependent* focus, rather than user-dependent.

2.8 Conclusion

In this chapter we have presented a scalable, language-independent technique for determining which query strings submitted to a Web search engine would benefit from automatic geo-localization. Using data from a search query log, we have shown that straightforward query tagging combined with an appropriate set of features and a standard supervised classifier can achieve up to 85% precision. A meta-classifier comprised of three conventional classifiers in a majority voting scheme performs even better, achieving 94% precision in cross-validation experiments.

We limited our experiments to identifying explicit locations within the United States, such as city and state names. This could reasonably be expanded, however, to include other locale data, such as specific sites or landmarks (e.g. "hotel near the Eiffel Tower").

CHAPTER 3

Ambiguous Queries

3.1 Introduction

Web search engines typically display a linear list of results for a user query, ranked by numerous factors such as relevance to the search terms and overall popularity. Search queries, however, are often underspecified, ambiguous, or multifaceted [San08, SLN09], and the search engine must find suitable ways to cope with them. The query "virus" could refer to, for example, a computer virus or a biological virus, and it is nearly impossible to know which meaning the user intended. With an ambiguous query, a few interpretations often dominate the top results, leaving less popular aspects uncovered. Users interested in less prevalent meanings encounter difficulty finding relevant documents.

In this chapter we study how a search engine can better serve users when they encounter an ambiguous query. Studies on search diversification aim to address this problem by introducing a diverse set of pages into search results [ZCL03, CK06, AGH09, ZWT09, WZ09]. Common to a majority of prior research, however, is the "single relevant document assumption." In fact some proposed approaches are provably optimal for various retrieval metrics under the assumption a user requires only *one* relevant document from their intended subtopic. This assumption is an over simplification. Many users will not be satisfied with only one relevant document, particularly for *informational* queries, and a search di-

versification strategy must properly account for them.

We focus our work on the problem of diversifying search results for informational queries. Improving the results for informational queries will significantly improve the search experience for many users because they tend to spend a disproportionate amount of time on informational queries. That is, navigational queries result in short interactions because the user already has a particular website in mind and simply uses the search engine as a pseudo-bookmark to locate the URL. For informational queries, however, the exact documents of interest are not known in advance. Users typically inspect the results for an informational query more in-depth, carefully exploring many pages in the result set [LLC05]. Optimizing these queries will reduce the burden placed on the user by helping them find a sufficient number of relevant documents more quickly.

The distinct search behavior for informational queries dictates the following modeling requirements:

1. Users often need more than one document to satisfy their information need, so the diversification model should properly account for users who need multiple relevant documents.

2. Ambiguous queries often have several potential subtopics. While a user tends to have one particular subtopic in mind, that subtopic is not known by the search engine.

3. The content of each document also tends to focus on only one of the possible subtopics, but the search engine lacks explicit topic classification for the majority of documents.

In this chapter we present a model that accounts for the above requirements for informational queries and define a measure of user satisfaction with respect to

that model. We then present an algorithm which introduces diversity into search results for informational queries such that we maximize the number of users who are able to find a sufficient number of documents related to their intended subtopic. We evaluate our algorithm against commercial search engine results and a recently proposed diversification algorithm [AGH09] on our proposed *expected hits* metric, as well as traditional metrics designed under single relevant document assumptions.

3.2 Diversification Model Overview

Given an ambiguous query, our goal is to select the set of documents which will satisfy the majority of users. Commercial search engines frequently return homogeneous document sets for such queries, which is sub-optimal in most cases. We therefore study ambiguous queries as a search diversification problem, with the goal of introducing diversity by identifying the relevant subtopics for an ambiguous query and using the probability of user interest in each of those subtopics to produce a document ranking which increases the likelihood an average user finds sufficient relevant documents.

We concentrate on informational queries, where users often require more than one relevant document. Our model takes probabilistic information about query intent, relevance of documents to the possible query subtopics, as well as the number of pertinent documents a user requires into consideration, and assumes these factors to be independent. By considering query intent likelihood, we are able to identify which subtopics are most important to the users. Document categorization probabilities help estimate how likely a document is to satisfy a particular subtopic. Estimating how many relevant documents a user will require enables us to weigh the expected benefits of providing additional documents from

already represented subtopics against exploring less covered subtopics. The assumption that users often require *multiple* documents relevant to their intended subtopic breaks with traditional work in search diversification.

Each of the necessary distributions are discussed further in the subsequent sections, followed by the definition of our goal metric.

3.2.1 Relevant Document Requirements

For informational queries it is important to consider how many relevant documents a user will visit. For example, if most users want to see 10 relevant documents, diversifying the results in the top 10 may actually lower the satisfaction for many users. The number of documents j a user requires to satisfy their need, however, is often relatively small. Showing a user more than j relevant documents is generally unnecessary. We model j as a distribution over the number of relevant documents a user requires: user U is expected to require j documents related to their subtopic of interest with probability $\Pr(J = j|U)$ for $j > 0$.

3.2.2 User Intent

User intent represents the likelihood an average user is interested in a particular subtopic of an ambiguous query. The user intent probability distribution is important for determining the relative importance of each subtopic. In our model, a user issues a search query for an ambiguous topic T which has m subtopics $T_1, T_2, \cdots, T_m$. For a given user U who queries for topic T, we consider a distribution over subtopics of interest to U: U is interested in subtopic T_i with probability $\Pr(T_i|U)$.

3.2.3 Document Categorization

Web search engines perform quite well at retrieving documents relevant to query terms. To select a diverse set of documents for an ambiguous query, however, first requires determining which subtopic(s) each document belongs to. Automatic classification is a difficult problem, and manual classification of documents is infeasible on a Web scale. Accurate document categorization is also important, as it tells us, probabilistically, which subtopic(s) a particular document satisfies. We model document categorization as a probability distribution. For a document d which is relevant to topic T, we assume a distribution over the subtopics: d is relevant to T_i with probability $\Pr(T_i|d)$.

3.2.4 Objectives

Given the probability distributions $\Pr(J|U)$, $\Pr(T_i|U)$, and $\Pr(T_i|D)$, the absence of any additional contextualizing information from the user, and a choice of any n documents to display, our task is to select documents such that we maximize the likelihood of user satisfaction.

To be clear about our objective, we must first define "user satisfaction". The simplest satisfaction measurement could be binary: a user either does or does not find as many documents as they desired from their intended subtopic. While it is possible to define a goal function and optimize for such criteria, this model does not seem to adequately reflect the real world. If a user wants five relevant documents, but only finds four to click on, they are likely still partially satisfied. We therefore define our objective in terms of *hits*, where a *hit* constitutes a click on a document which satisfies the subtopic the user is interested in. We then achieve our goal of optimal user satisfaction by maximizing the expected number of *hits* for the average user.

Consider a simplified example where a user issues the query *virus*. Assume they are interested in biological viruses, and 3 of the returned documents R are about biological viruses. Using the required-documents distribution $\Pr(J|U)$ we can calculate how many documents the user is expected to click on. If the user is interested in one, two, or three relevant documents, they are expected to click on as many. If they are interested in more than three documents, they can only click on the three that are displayed. We thus compute the expected number of hits H for a user U and set of documents R as:

$$E(H|R,U) = 1 \cdot \Pr(J=1|U) + 2 \cdot \Pr(J=2|U) + 3 \cdot \sum_{j=3}^{|R|} \Pr(J=j|U)$$

The above example shows how, given $\Pr(J|U)$, we can compute the expected number of hits for a set of documents when user intent and document categorizations are known. In reality these are not known values, but rather probability distributions. In the next section we will show how these distributions factor in to the model and present our algorithm for selecting a set of results R such that we maximize the expected number of hits.

3.3 Diversification Model

The general approach we take is to successively select documents, at each step choosing the document which adds the maximum additional expected hits. If our goal were to return at least one relevant result, this document would most likely come from a subtopic not yet covered. In our more general model, however, this is not always the case, as we may benefit more users by returning additional documents from a popular subtopic.

To determine how to best select documents, we must examine the effects of the probability distributions discussed in Section 3.2 on the expected number of

hits. We begin by analyzing two simplified cases of those distributions. First, we will assume perfect knowledge of user intent. Second, we will assume perfect document classification.

3.3.1 Perfect Knowledge of User Intent

The first case we examine is when we know exactly which subtopic T_i a user is interested in but document classification is probabilistic. To calculate the expected number of hits for a set of documents when we know the user intent, we must consider how many documents j the user requires, and how many of the documents presented are relevant, denoted as k. A user will click on at most j documents, so returning more than j is unnecessary. Likewise, a user will see at most k relevant documents, and thus can click on no more than k.

When the subtopic of interest to user U is known, we can compute the expected number of hits H for a set of n documents R as:

$$E(H|R,U) = \sum_{j=1}^{n} \Pr(J = j|U) \sum_{k=1}^{n} \Pr(K_i = k|R) \min(j,k) \tag{3.1}$$

In Equation 3.1, K_i is defined as the event that k documents in R belong to T_i. To compute this probability, we begin by defining the probability that no documents from R satisfy T_i as:

$$\Pr(K_i = 0|R) = \prod_{r=1}^{n} (1 - \Pr(T_i|d_r))$$

In the general case where a user requires k relevant documents, we can expand this equation to:

$$\begin{aligned} \Pr(K_i = k|R) = {} & \Pr(T_i|d_1) \Pr(K_i = k-1|R \setminus \{d_1\}) \\ & + (1 - \Pr(T_i|d_1)) \Pr(K_i = k|R \setminus \{d_1\}) \end{aligned}$$

From Equation 3.1, $\Pr(J|U)$ is independent of which subtopic the user is interested in, and thus only $\Pr(K_i|R)$ will be affected by the choice of documents. Since $\Pr(K_i = k|R)$ is the only term in the equation dependent on the selected documents, and the user is only interested in subtopic T_i, we can maximize the the expected number of hits by selecting the documents with the highest $\Pr(T_i|D)$ values, that is, by maximizing $\Pr(K_i = n|R)$. Under these conditions, our strategy for selecting documents is similar to the greedy approach for optimizing k-call presented by Chen and Karger [CK06], using $\Pr(T_i|D)$ to select the documents most likely related to T_i.

3.3.2 Perfect Document Classification

We next make the assumption that each document is classified into a single subtopic category, but user intent is unknown. In terms of the probability distributions described in Section 3.2, perfect classification means D is divided into non-overlapping subsets $D_1, D_2, \cdots D_m$ such that for each subtopic T_i, $\Pr(T_i|d \in D_i) = 1$ and $\forall_{j \neq i} \Pr(T_j|d \in D_i) = 0$.

In this case, we study how to combine user intent and relevant document requirement distributions to best allocate documents from subtopics and maximize user satisfaction. We start by again defining the number of documents selected from subtopic T_i as K_i and enforce the condition that for the m subtopics of T, $\sum_{i=1}^{m} K_i = n$. As in the previous case, a user will click on up to j documents from subtopic T_i, and can click on at most K_i documents if $K_i < j$.

When each document is classified to perfectly satisfy a single subtopic, we can calculate the expected number of hits for an average user U and set of n

documents R with the following equation:

$$E(H|R,U) = \sum_{j=1}^{n} \sum_{i=1}^{m} \Pr(T_i|U) \Pr(J = j|U) \min(j, K_i) \tag{3.2}$$

3.3.2.1 Solving For K

When we know exactly which subtopic each document belongs to, our main task becomes deciding how many documents from each subtopic should be included in the results. That is, we need to pick the set of $\{K_i\}$ values which will maximize the expected number of hits.

Given all the possible values for $\{K_i\}$, we can calculate the expected hits of each and choose an optimal solution. With n documents to choose from m subtopics, the number of combinations of $\{K_i\}$ values which satisfy the requirement $\sum_{i=1}^{m} K_i = n$ is $\binom{n+m-1}{n}$, making it infeasible to consider all possible combinations for a query. We can greatly reduce the search space, however, as many combinations are clearly not optimal. Allocating all documents to the least probable subtopic, for example, will not result in the maximum number of hits. Intuitively, an optimal solution should contain at least as many documents from the most probable subtopic as a less popular one. We formalize this notion with the following Proposition:

Proposition 3.3.1 *Without loss of generality, label the subtopics of topic T as $T_1, T_2, \cdots, T_m$ such that $\Pr(T_1|U) \geq \Pr(T_2|U) \geq \cdots \geq \Pr(T_m|U)$. Then an optimal solution to Equation 3.2 satisfies the following properties:*

- $\sum_{j=1}^{n} \Pr(J = j|U) = 1$
- $\sum_{i=1}^{m} K_i = n$

- $K_1 \geq K_2 \geq \cdots \geq K_m$

Proof Assume an initial set of documents R with K values $\{K_1, K_2, \cdots, K_m\}$ such that $\sum_{i=1}^{m} K_i = n$ and $K_x < K_y$ for some $x < y$, with expected number of hits $E(H|R,U)$ as defined in Equation 3.2. Then we can construct a document set $\hat{R}$ with $\hat{K} = \{K_1, \cdots, K_x + 1, \cdots, K_y - 1, \cdots K_m\}$ with expected hits:

$$\begin{aligned}
E(H|\hat{R},U) &= E(H|R,U) \\
&\quad + (\Pr(T_x|U) \sum_{j=K_x+1}^{n} \Pr(J=j|U)) \\
&\quad - (\Pr(T_y|U) \sum_{j=K_y}^{n} \Pr(J=j|U)) \\
&\geq E(H|R,U) + (\Pr(T_x|U) - \Pr(T_y|U)) \sum_{j=K_y}^{n} \Pr(J=j|U) \\
&\geq E(H|R,U)
\end{aligned}$$

3.3.2.2 Document Selection

In practice it is not necessary to enumerate and test all possible $\{K_i\}$ values, as we can optimize for Equation 3.2 directly. We select the documents to return using an algorithm which factors in both $\Pr(T_i|U)$ and $\Pr(J|U)$ while adhering to Proposition 3.3.1, and update K_i after each selection accordingly.

To choose each successive document, *KnownClassification* takes a greedy approach. The algorithm first determines which subtopic will provide the maximum marginal benefit to the average user. The marginal utility of a subtopic is the expected increase in hits produced by adding another document from it, and is the product of the user interest in the subtopic $\Pr(T_i|U)$ and the probability that users will want another document from that subtopic $\Pr(J > K_i|U)$. Once the

Algorithm *KnownClassification*

(∗ Rank documents to maximize Equation 3.2 ∗)

1. $R \leftarrow \emptyset$
2. $D \leftarrow$All relevant documents
3. $K_1 = K_2 ... = K_m = 0$
4. **while** $|R| < n$
5. $\quad i \leftarrow \text{ARGMAX}(\Pr(T_i|U) \Pr(J > K_i|U))$
6. $\quad K_i \leftarrow K_i + 1$
7. $\quad R \leftarrow R \cup \text{NextDocument}(D_i, R)$

next subtopic is chosen, a search engine may select the next document to return from $D_i \setminus R$ using its standard ranking functions.

3.4 Complete Model

We now eliminate the simplifying assumptions and discuss how to compute the expected hits when neither document classifications nor user intent are perfectly known. With user intent uncertain, we need to calculate the expected hits probabilistically over all of the possible subtopics instead of only a single, known T_i from Equation 3.1. From Equation 3.2 we can no longer say the user will click on $\min(j, K_i)$ documents, as we have no guarantees on the number of documents which actually satisfy subtopic T_i. Instead, we expect the user to click on $\min(j, k)$ documents, based on the probability that k relevant documents are available to the user.

Combining the two simplified equations and making use of all three probability

distributions, the equation for expected number of hits becomes:

$$E(H|R,U) = \sum_{j=1}^{n} \sum_{i=1}^{m} \Pr(T_i|U) \Pr(J = j|U) \sum_{k=1}^{n} \Pr(K_i = k|R) \min(j,k) \quad (3.3)$$

Algorithm *Diversity-IQ*

(∗ Rank documents to maximize Equation 3.3 ∗)

1. $R \leftarrow \emptyset$
2. $D \leftarrow$All relevant documents
3. **while** $|R| < n$
4. $\quad d \leftarrow$ARGMAX($\Delta E(d|R,D)$)
5. $\quad R \leftarrow R \cup \{\text{d}\}$
6. $\quad D \leftarrow D \setminus \{\text{d}\}$

Diversity-IQ outlines how to select the set of documents R such that we maximize the expected number of hits for an ambiguous informational query. We adopt the greedy approach of *KnownClassification*, selecting each successive document by determining which will maximize the increase in expected hits given the documents already returned.

The ΔE computation for a document is dependent on several factors, including its subtopic scores, the user interest in those subtopics, and the conditional probabilities of how many documents from each subtopic are already included in R. After a document is selected and added to R, the $\Pr(K_i|R)$ values from Equation 3.3 can be updated and used in the next iteration. Thus we have an overall computational complexity of $O(|R| \cdot |D| \cdot m)$ for choosing each successive document, or $O(n^2 \cdot |D| \cdot m)$ for re-ranking the top n documents.

3.5 Comparison With IA-Select

In this section we briefly go over the related work on search diversification by Agrawal et al. [AGH09] to better understand when prior work may perform sub-optimally, and how our approach may overcome such scenarios.

3.5.1 Overview of IA-Select

Agrawal et al. investigate the problem of ambiguous queries with the overall objective of maximizing the probability that an average user finds *at least one* relevant document in the top n search results. Their model assumes an explicit taxonomy of subtopics is available, and both documents and queries may fall into multiple subtopics. Queries belong to a set of subtopics with a known probability distribution, which effectively represents the user intent for a given query $\Pr(T_i|U)$. Likewise, documents belong to a set of subtopics, and the relevance to each subtopic is measured probabilistically, much like $\Pr(T_i|D)$.

Given this model and set of distributions, they formulate the *Diversify* function, which measures the probability that a set of n documents satisfies an "average" user for an ambiguous query. The objective to select the set of documents which maximizes this probability is proven to be NP-Hard, and the authors propose the *IA-Select* algorithm as an approximation, which is shown to produce an optimal solution to *Diversify* when every document belongs to a single subtopic.

Key to their algorithm is the notion of a conditional probability of subtopics, $U(T_i|R)$, which measures the probability that the user is still interested in subtopic T_i given the documents already chosen in R. The conditional probability of each subtopic is initialized to the user intent probability $\Pr(T_i|U)$. The algorithm successively selects documents which have the highest marginal utility, computed

for each document as the sum, over each subtopic, of the subtopic's conditional probability and the document's score for that subtopic:

$$g(d|R) = \sum_{i=1}^{m} \Pr(T_i|d)U(T_i|R)$$

After a document d is selected, the conditional probability of each subtopic is updated to reflect the inclusion of d in R using Bayes' theorem:

$$\forall i : U(T_i|R) = (1 - \Pr(T_i|d))U(T_i|R \setminus \{d\}) \tag{3.4}$$

3.5.2 Observed Limitations of IA-Select

In our experiments with *IA-Select*, we observed the algorithm often selects one document from each subtopic, in the order of subtopic popularity, and then degenerates into seemingly random document selection. We believe this behavior is sub-optimal. Even if every subtopic is represented once in the results, an average user will want to see additional documents from more popular subtopics if there is room.

From our investigation, we find that this behavior is due to the following limitation. When deciding which document to select next, *IA-Select* uses the conditional probability $U(T_i|R)$, which measures the likelihood that the user is still interested in subtopic T_i given the documents already selected in set R. *IA-Select* assumes that the user is no longer very interested in subtopic T_i once at least one document believed to satisfy T_i is present in R, meaning $U(T_i|R)$ becomes very small.

When *IA-Select* is used with any document classification function which assigns subtopic scores approaching 1.0, the Bayesian update step in Equation 3.4 is problematic. To illustrate the issue more clearly, consider an extreme case where a document is classified to "perfectly" belong to any subtopic ($\Pr(T_i|d) = 1$). In

that case, the subtopic will have its conditional probability set to zero. That is, if even one document from each subtopic has such a score, every conditional utility value will be set to zero, and the algorithm is reduced to random selection. Note that this behavior is not limited to the extreme case when $\Pr(T_i|d) = 1$. As long as the $\Pr(T_i|D)$ values are sufficiently high, all conditional subtopic probabilities will quickly become very small, and the algorithm exhibits similar behavior.

Zero-utility is particularly problematic if we consider that selecting multiple documents from a subtopic may be beneficial, which may be the case even in simple situations such as a query having fewer subtopics than document "slots" to fill ($m < n$). We avoid the zero-utility problem by computing the marginal benefit of a subtopic in terms of the probability that a user wants additional documents from it, which depends on $\Pr(J|U)$ and $\Pr(K_i|R)$. As long as $\Pr(J|U) > 0$, each subtopic will always have non-zero utility.

To illustrate the issue more clearly and accentuate how our algorithm avoids the zero-utility problem, we will walk through a simple example. In this example we use binary classification scores for clarity and to underscore the potential problems with *IA-Select* only. As we will see later in our experimental section, *IA-Select* exhibits similar behavior, to a lesser degree, even under widely used probabilistic classifiers which may assign any value between 0 and 1.

3.5.3 Descriptive Example

Assume two subtopics T_1 and T_2, with two documents classified into each subtopic. Our example will use the following probabilities and subtopic scores:

- $\Pr(T_1|U) = 0.7$ and $\Pr(T_2|U) = 0.3$.
- $\Pr(J|U) = (0.6, 0.3, 0.1)$.

- $D = \{d_1 = d_2 = (1.0, 0.0), d_3 = d_4 = (0.0, 1.0)\}$
- $n = 3$.

3.5.3.1 *Diversity-IQ*

To choose the first document, *Diversity-IQ* computes the marginal utility of each document:

$$\Delta E(d_1|\emptyset) = \Delta E(d_2|\emptyset) = 0.7 \sum_{j=1}^{3} \Pr(J = j|U) = 0.7$$

$$\Delta E(d_3|\emptyset) = \Delta E(d_4|\emptyset) = 0.3 \sum_{j=1}^{3} \Pr(J = j|U) = 0.3$$

The first document selected is chosen arbitrarily between $\{d_1, d_2\}$. To choose the second document, we compute the marginal utility of each remaining document:

$$\Delta E(d_2|\{d_1\}) = 0.7 \sum_{j=2}^{3} \Pr(J = j|U) = 0.28$$

$$\Delta E(d_3|\{d_1\}) = \Delta E(d_4|\{d_1\}) = 0.3$$

As d_3 and d_4 both provide the same increase in expected hits, we again choose arbitrarily between them. Thus we have $R = \{d_1, d_3\}$ after the first two iterations. To choose the third document, we again compute the marginal utility of the remaining documents:

$$\Delta E(d_2|\{d_1, d_3\}) = 0.7 \sum_{j=2}^{3} \Pr(J = j|U) = 0.28$$

$$\Delta E(d_4|\{d_1, d_3\}) = 0.3 \sum_{j=2}^{3} \Pr(J = j|U) = 0.12$$

Since d_2 has a higher marginal utility than d_4, it is added to the result set, for a final ranking $R = \{d_1, d_3, d_2\}$ with expected hits $E(H|R, U) = 1.28$.

3.5.3.2 IA-Select

For *IA-Select*, we initialize the utility of each subtopic to the user intent probabilities and compute the marginal utility of each document:

$$g(d_1|\emptyset) = g(d_2|\emptyset) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.7$$
$$g(d_3|\emptyset) = g(d_4|\emptyset) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.3$$

After choosing arbitrarily between $\{d_1, d_2\}$, we update the conditional probability of the subtopics:

$$U(T_1|\{d_1\}) = (1 - \Pr(T_1|d_1))U(T_1|\emptyset) = 0.0$$
$$U(T_2|\{d_1\}) = (1 - \Pr(T_2|d_1))U(T_2|\emptyset) = 0.3$$

We recompute the marginal utility of each document:

$$g(d_2|\{d_1\}) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.0$$
$$g(d_3|\{d_1\}) = g(d_4|\{d_1\}) = \sum_{i=1}^{2} \Pr(T_i|d)U(T_i|R) = 0.3$$

Again, we choose arbitrarily between $\{d_3, d_4\}$ and update the conditional probability for each subtopic:

$$U(T_1|\{d_1, d_3\}) = (1 - \Pr(T_1|d_3))U(T_1|\{d_1\}) = 0.0$$
$$U(T_2|\{d_1, d_3\}) = (1 - \Pr(T_2|d_3))U(T_2|\{d_1\}) = 0.0$$

At this point, we still need to select a third document ($n = 3$), but the conditional utility of each subtopic is zero, meaning the marginal utility of every document will be zero. Intuitively, we would expect the more probable subtopic

to be a better choice for the "average" user in this situation, but *IA-Select* will randomly choose between $\{d_2, d_4\}$. Note that there is a substantial difference in the expected hits depending on which we choose: d_2 will increase the expected hits by 0.28, while d_4 by only 0.12.

3.5.4 Discussion

One may contend that the outlined issues can easily be patched by smoothing or enforcing a limit on the maximum score assigned to any particular subtopic. In our evaluations we will show that, while placing such arbitrary limits on subtopic scores can improve *IA-Select's* performance on the expected hits metric to a certain degree, it still exhibits similar behavior, and the improvements come at the cost of degraded performance on other metrics.

It is also worth noting that Equation 3.3 is, in fact, a generalization of the *Diversify* goal function. That is, if we make the assumption that all users require exactly one document (setting $\Pr(J = 1|U) = 1$), *Diversity-IQ* will yield the same ranking as *IA-Select*.

3.6 Distribution Measurements

Our algorithm requires three distributions which describe (1) the number of relevant documents a user is expected to require, (2) the probability of user intent in each subtopic, and (3) the probability a document satisfies each subtopic. Given the broad range of possible queries and the number of documents on the Web, automatic methods for approximating these distributions are necessary for a real world deployment. In this section we suggest techniques to approximate them using data sources available to Web search engines. In Section 3.7 we specify

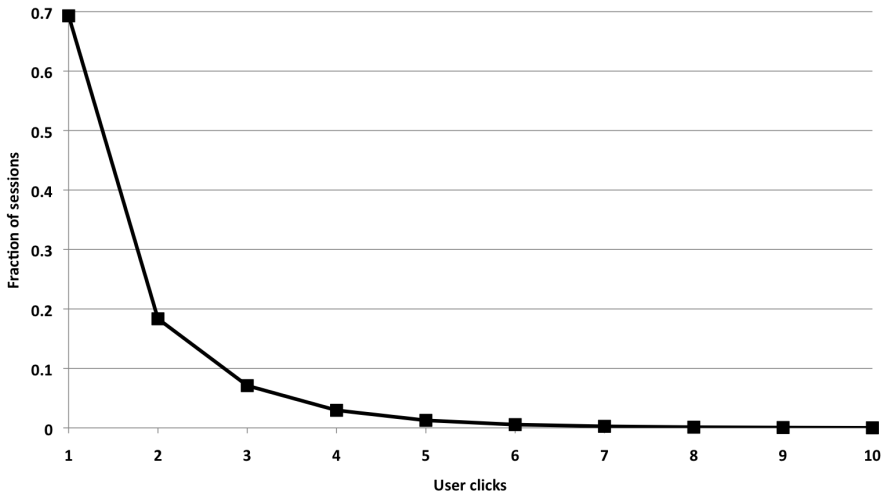

Figure 3.1: Distribution of clicks per query

which data sources are used for each experiment.

3.6.1 Measuring Document Requirements

Knowing the number of relevant documents a user is expected to require is necessary to determine how much diversity we can introduce in the results without harming the hit-rate for popular subtopics. One method to approximate this distribution is using clickthrough data from query logs. Figure 3.1 shows the number of clickthroughs for each query session with at least one click from a locally collected search query log [QLC05]. We observe that other publicly available query logs show a similar distribution. In our log, users clicked on an average of 1.52 results for queries with at least one clickthrough. Other studies of web search logs report an average of 3.18 clicks-per-query [OKK02] when empty sessions are removed.

3.6.2 Measuring User Intent

The user intent distribution measures the probability that an average user is interested in a particular subtopic for a given query. We have shown in Section 3.3.2.1 that if subtopic T_i is *more likely* than subtopic T_j, then showing the user at least as many documents from T_i as T_j is a necessary condition for optimizing the expected number of hits. Therefore an accurate estimate of user interest in each subtopic is important. Possible sources for this information include:

- The frequency of popular query refinements for ambiguous queries [RD06].
- The clickthrough history for documents returned by an ambiguous query.
- Frequency of subtopic queries, which can be measured using information about search volume and trends available from commercial search engines such as Google [Goo] and Bing [Bin].

3.6.3 Measuring Document Categorization

The document categorization distribution tells us which of the m subtopics a particular document belongs to. Unsupervised document classification techniques often require an estimate for m and a sufficiently large collection of relevant documents. We look at these issues next.

3.6.3.1 Subtopic Estimation

We investigated two sources for discovering the subtopics for a given ambiguous query: (1) WordNet [Fel98], and (2) Wikipedia [Wik]. WordNet is a popular lexical database which includes term relationships. Unfortunately, data for queries such as movies, song titles, and proper nouns are sparse in WordNet. The second

source we examined is Wikipedia. Among the millions of articles on Wikipedia are over 60,000 disambiguation pages for the English language. These pages list several possible meanings of term, covering a wider range of entity types.

Other approaches for identifying subtopics include mining popular query refinements [RD06] and using categories from the Open Directory Project [Pro].

3.6.3.2 Document Classification

Radlinski and Dumais show that homogeneity in the top results generally hinders the effectiveness of personalization [RD06]. Without a sufficient number of documents from each subtopic, unsupervised classification techniques will be unable to generate meaningful topics. In our experiments we therefore opt for a metasearch strategy. We form a collection of documents for an ambiguous query by issuing each relevant Wikipedia subtopic page title as a search query. We merge the top 200 results from each subtopic query to form a single document set.

Given a set of m subtopics T and collection of documents D, a document classification function $C(d, T)$ assigns a probability score $\Pr(T_i|d)$ for each $T_i \in T$, such that $\sum_{i=1}^{m} \Pr(T_i|d) = 1$. We consider two such classification functions: (1) *query-based* classification, and (2) Latent Dirichlet Allocation (LDA) [BNJ03].

Query-based classification uses knowledge of which subtopic queries returned each document. For each of the subtopic queries that returned a document d in its top 200 results, we compute a score $S_i(d) = cos(d^c, W(T_i))$, where d^c and $W(T_i)$ are vector space model representations of the text from the search result snippet and the Wikipedia page for subtopic T_i, respectively, and *cos* is the cosine similarity function. We normalize these scores to assign a final score to

each subtopic for a document as:

$$\Pr(T_i|d) = \frac{S_i(d)}{\sum_{j=1}^{m} S_j(d)}$$

The second document classification method we consider is Latent Dirichlet Allocation (LDA). LDA requires a set of documents, a number of topics m, and two hyperparameters α and β which control smoothing of Dirichlet priors for topics and words. In typical applications, $0 < \alpha < 1$ and β is set to 0.1 or 0.01 [SG07]. We construct an LDA topic model for each document set and assign $\Pr(T_i|d)$ directly from the resulting document-topic (θ) distribution.

3.7 Evaluation

We conducted several experiments to assess the overall effectiveness of *Diversity-IQ*. The evaluations include an analysis of our objective of maximizing the expected number of hits, as well as comparisons using established subtopic retrieval metrics. For each metric, we compare our *Diversity-IQ* algorithm against the *IA-Select* algorithm [AGH09] as well as the original ranking returned by a commercial Web search engine (SE).

3.7.1 Query Set

One of the difficulties in evaluating a system designed to introduce diversity is the lack of standard testing data. Evaluating diversification requires a set of ambiguous queries, and until recently, no benchmark query sets or relevance judgements exist explicitly for the task of diversification research. TREC added a diversity task to the Web track beginning in 2009. The data includes 50 queries, each with a set of selected subtopic aspects. Unfortunately, this benchmark dataset and evaluation criteria were designed under the single relevant document

assumption, and so it is difficult to adapt them to our multi-document setting.

Although techniques exist to identify ambiguous queries [SLN09] which would be beneficial in a real world deployment, we are primarily concerned with evaluating the performance benefits of our algorithm, and thus we opted for a simpler approach to form a testing set. We generated a set of ambiguous queries using a small search log with a few hundred thousand entries collected from our local network. A query is marked as ambiguous if a Wikipedia disambiguation page exists for the terms. We randomly selected 50 queries from this candidate set for our evaluations.

3.7.2 Probability Distributions

Figure 3.1 shows the clickthrough distribution for *all* queries, but studies indicate that navigational queries account for anywhere from 10-25% of Web searches [Bro02, RL04] and typically result in a single click [LLC05]. Removing these queries from the query log would produce a more accurate distribution for our algorithm, but automatically classifying queries as informational or navigational is a difficult task. To avoid unfairly penalizing our algorithm with a clickthrough distribution containing navigational queries, we approximate the distribution of how many relevant documents a user will require using the geometric series $\Pr(J = j|U) = 2^{-j}$, which represents an average of 2 clicks-per-query, displays an exponential decay characteristic like Figure 3.1, and has the property $\lim_{n\to\infty} \sum_{j=1}^{n} \Pr(J = j|U) = 1$, which conforms with the conditions of Proposition 3.3.1.

To measure the user intent distribution $\Pr(T_i|U)$ for our experiments, we conducted a survey using Amazon's Mechanical Turk [Tur]. For each query we asked 10 survey participants to select all of the subtopics they associate with the

query from the available choices. To keep the task manageable, we limited our study to queries with at most 20 subtopics, with an average of 8.5 subtopics per query. To ensure all subtopics were considered, those which received no votes were assigned a small non-zero value of 0.01.

Unless otherwise noted, the document-subtopic probability scores $\Pr(T_i|D)$ were assigned using the GibbsLDA++ [PN] implementation of LDA (see Section 3.6.3.2). Parameters were set at $\alpha = 0.2$ and $\beta = 0.1$, based on values found to work well for text collections [GS04].

3.7.3 Expected Hits

We analyzed *Diversity-IQ* with our overall goal metric of maximizing the expected number of hits. For each test query we compute the expected number of hits for each ranking strategy over increasing values of n using Equation 3.3. A majority of users do not look beyond the first result page [JS06], making efficiency of the top documents particularly important. We therefore limit our evaluation to the top 10 results, as commercial search engines typically show 10 results per page.

Figures 3.2 and 3.3 show the mean expected hits computed over the test query set for the top 10 results with the three ranking approaches. Figure 3.2 assigns $\Pr(T_i|D)$ using the subquery-based classification method, while Figure 3.3 uses LDA to assign subtopic scores. In both cases, the expected hits from the top document is comparable, as providing at least one document from the most probable subtopic is generally the initial strategy taken by both our algorithm and *IA-Select*. After the top few results, however, our algorithm may find additional benefits from providing additional documents from popular subtopics, and thus our algorithm tends to increase the expected hits more rapidly.

We measured the runtime performance of each algorithm on a 2.6 GHz Intel

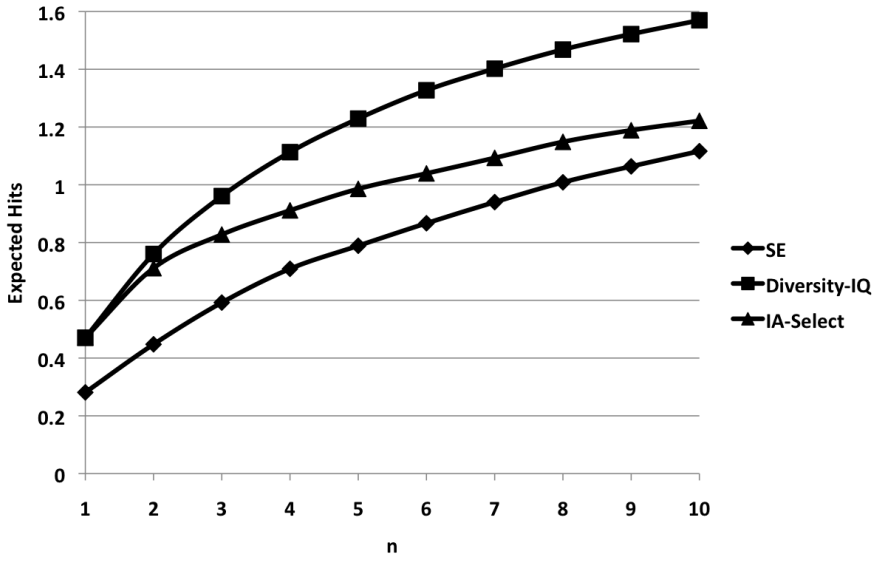

Figure 3.2: Expected hits (Query-based document classification)

Core 2 Duo CPU with 4 GB memory running Mac OS X 10.6. Implementations were written in Python 2.6. To select the top 10 results, *Diversity-IQ* required an average of 28.8ms, while *IA-Select* averaged 28.5ms.

3.7.3.1 Classification Score Range

We briefly look at how the range of average classification scores can effect the expected hits for both algorithms. For each query we performed LDA classification with varying values of α and β. We identify the subtopic with the highest individual score for each document and compute the average of these scores over all documents. Figure 3.4 plots this average "top" subtopic score against the corresponding expected hits for each algorithm. The figure shows that *Diversity-IQ* outperforms *IA-Select* on expected hits regardless of the classification scores, and that as potential subtopic scores approach 0.7, *IA-Select* suffers a significant drop in performance.

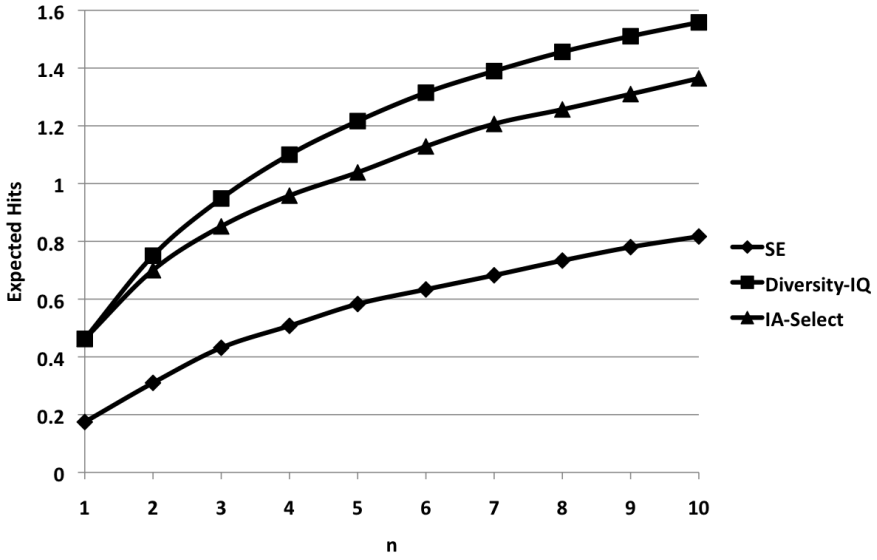

Figure 3.3: Expected hits (LDA document classification)

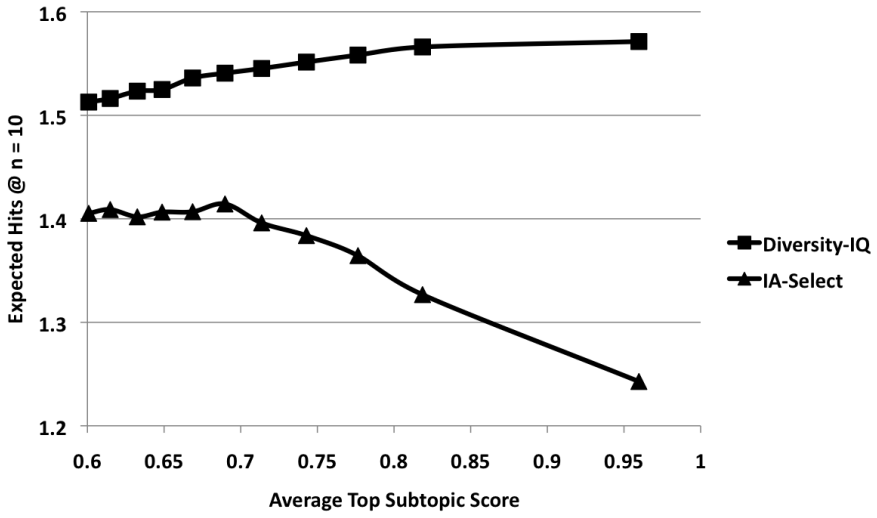

Figure 3.4: Effect of average subtopic scores on expected hits

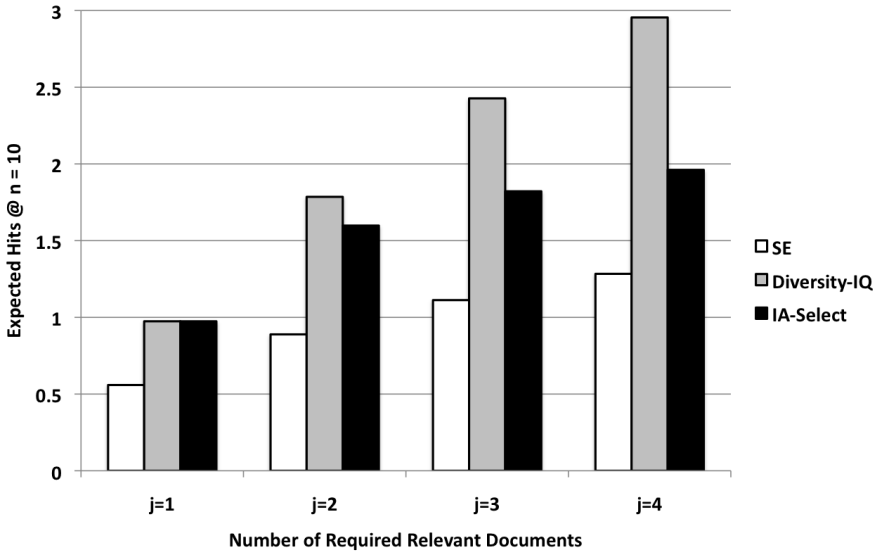

Figure 3.5: Effect of varying the number of required documents

3.7.3.2 Requiring Multiple Documents

We next study the effects of $\Pr(J|U)$ on expected hits, and in particular performance as the number of relevant documents a user is expected to require increases. Figure 3.5 shows the expected hits for $n = 10$ as we vary the number of documents users are expected to require from 1 to 4. As we can see, for users who require only one relevant document ($j = 1$), our algorithms have equal performance. In all cases where users want more than one document, however, *Diversity-IQ* outperforms *IA-Select*. As expected, we can see that our algorithm's relative performance improves as users are expected to require additional documents.

3.7.4 Single Document Metrics

Having demonstrated the performance advantages of our algorithm with respect to the more general model, we turn our attention to metrics based on returning *at least one* relevant document. As our algorithm may find it beneficial to return

multiple documents from popular subtopics before any documents from unpopular subtopics, we expect an algorithm focused on returning at least one relevant document, such as *IA-Select*, to outperform ours on these metrics. Nonetheless, it is important to compare the algorithms on a level playing field and quantify the differences in these scenarios as well.

3.7.4.1 Subtopic Recall

Subtopic recall (S-recall) at rank N was defined by Zhai, Cohen, and Lafferty as the percentage of relevant subtopics covered by the top N documents [ZCL03]. Assuming all users want one relevant document and a uniform user intent probability distribution, S-recall serves as an indication of expected user satisfaction with the top N. S-recall requires *binary* relevance judgements: a document either does or does not satisfy a particular subtopic. To compute the S-recall for our evaluation we consider a document as satisfying a subtopic if its subtopic score is above a certain threshold, which we set at $\Pr(T_i|d) \geq 0.3$.

Figure 3.6 plots the average subtopic recall for our evaluation set. As expected, *IA-Select* outperforms our algorithm on S-recall for the highest ranked documents. Our algorithm, however, outperforms the original search engine ranking, and on average covers over one-half of the subtopics within the top 10 results.

3.7.4.2 MRR-IA

S-recall assumes all subtopics are equally important. In reality we know that certain subtopics are often considerably more likely than others. To evaluate the effectiveness of our algorithm identifying such subtopics and presenting them early, we consider the "intent aware" Mean Reciprocal Rank (MRR-IA) metric defined in [AGH09]. MRR-IA measures the traditional mean reciprocal rank over

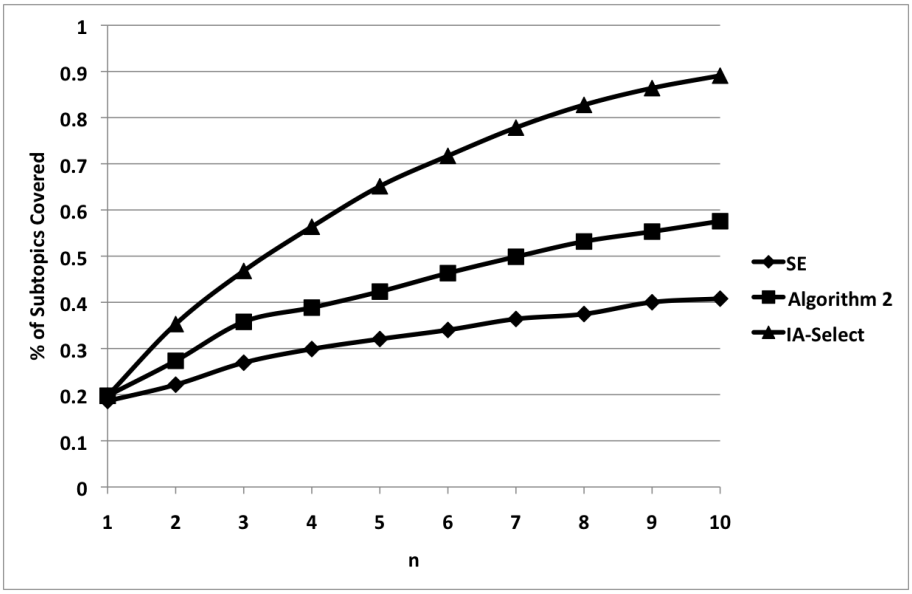

Figure 3.6: Subtopic recall

each subtopic, weighted by their probability of user intent. Again, we use the threshold 0.3 to determine whether or not a document satisfies a subtopic.

Figure 3.7 shows our algorithm outperforms the original search engine ranking and a small decrease in performance (approximately -6%) with respect to *IA-Select* at $n = 10$. This indicates that our algorithm is still able to identify the most probable subtopics and present at least one document from each early in the ranking, thus performing well for a majority of users even when we assume one relevant document is sufficient.

3.7.5 Smoothing IA-Select

As noted earlier, we can partially address the weakness in *IA-Select* by imposing limits on the maximum score assigned to any particular subtopic. We now evaluate the effects of varying that limit on expected hits, MRR-IA, and S-recall. For these experiments, we modified the Bayesian update step of *IA-Select*

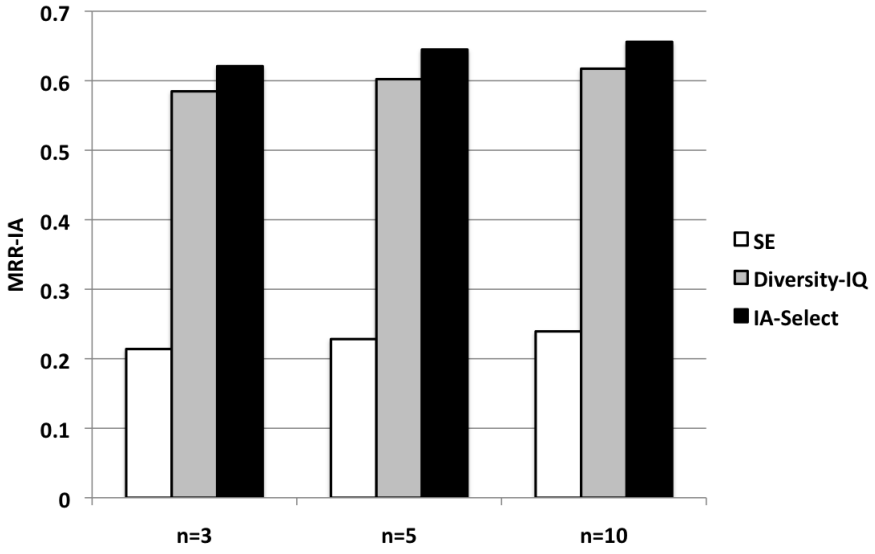

Figure 3.7: Intent-aware mean reciprocal rank (MRR-IA)

(shown in Equation 3.4) to multiply the conditional utility U of each subtopic by $(1-\min(\Pr(T_i|d), L))$, where L is the maximum allowable score. Figure 3.8 shows the effects of smoothing on IA-Select for various limits on the maximum subtopic scores. The general trend shows that, as we decrease the maximum allowable subtopic score, the expected hits increase as the other metrics decrease. It is unclear how to intelligently select a proper "smoothing" value for any particular number of relevant documents required.

3.8 Conclusion

In this chapter we presented an algorithm for diversifying search results when a Web search engine encounters ambiguous informational queries, where users often require multiple relevant documents. We described a model for user satisfaction with a set of search results, represented by the expected number of hits, or user clicks on relevant documents, in the top n. Our algorithm shows how, when faced

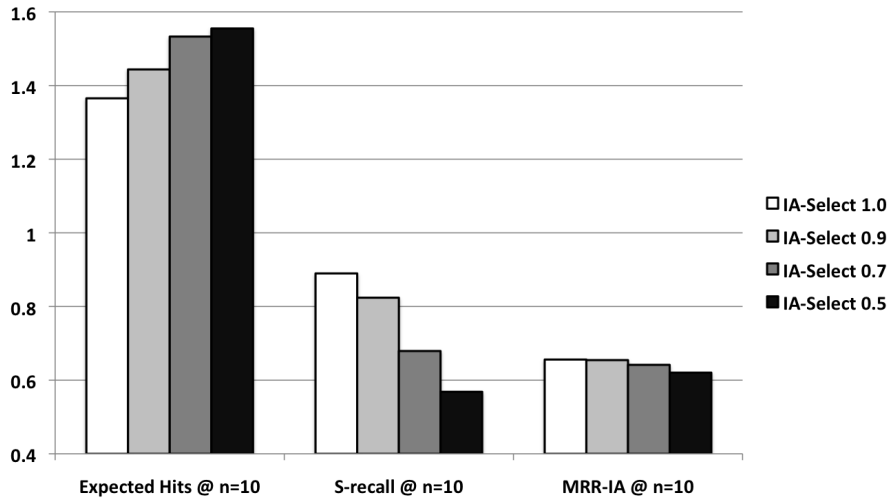

Figure 3.8: Effects of smoothing on IA-Select

with an ambiguous query, a search engine can use probabilistic knowledge of user intent, document classification, and how many relevant documents a user will require to return a document set which improves the probability of satisfaction for an average user.

Experiments show our *Diversity-IQ* algorithm improves expected user satisfaction for ambiguous informational queries, and helps overcome the limitations of earlier work by performing well regardless of the subtopic scores assigned by the document classification function.

3.9 Related Work

Search diversification as a strategy to better manage ambiguous queries has been studied in several contexts with many different approaches. Early techniques focused on the content of documents already selected, traditionally weighing between measures of query relevance and relative novelty of new documents. These

methods tend to produce diversity as a side effect of novelty and make no use of explicit knowledge of potential subtopics or user intent. Carbonell and Goldstein's work on Maximal Marginal Relevance (MMR) [CG98] is a classic example of such a strategy, which can be employed to re-rank documents and promote diversity.

Chen and Karger [CK06] use Bayesian retrieval models and condition selection of subsequent documents by making assumptions about the relevance of the previously retrieved documents. While their approach is capable of selecting anywhere between $0 < k \leq n$ relevant documents, they focus primarily on optimizing single document ($k = 1$) and perfect precision ($k = n$) scenarios. Their model, like MMR, does not explicitly consider user intent or document categorizations, making it difficult to prioritize more probable subtopics at the highest ranking positions. It is also unclear how their technique can best be applied to interleave documents from multiple subtopics into a single ranking when single document assumptions are removed.

Zhai, Cohen, and Lafferty [ZCL03] propose a framework which models dependent relevance and describe a generic greedy approach to ranking documents for subtopic retrieval. Their ranking strategy is based on a tradeoff between selecting documents of high value and minimizing cost, where documents which include relevant, previously uncovered information have higher value, and those that are irrelevant or repeat already seen information have a larger cost. With the goal of optimizing a ranking for their subtopic recall (S-recall) and subtopic precision (S-precision) metrics, their work implicitly assumes that a single document relevant to a category is sufficient for a user.

Wang and Zhu introduce an approach to diversification based on economic portfolio theory [ZWT09, WZ09]. Their models consider a "risk" tradeoff between

the expected relevance of a set of documents and correlation between them, modeled as the mean and variance. They demonstrate algorithms capable of a wide range of "risk preferences", though it is unclear how to choose the proper parameters to maximize their algorithm's performance under our proposed model.

Agrawal et al. introduce a model similar to ours in [AGH09], where their objective is to maximize the probability an average user finds *at least one* useful result. Under assumptions of probabilistic query intent and document categorization, they present a proof showing the selection of documents which optimize against that criteria is NP-hard, and offer an approximation algorithm with a bounded error from the optimal solution under certain assumptions. They also show their algorithm is optimal when all documents belong to a single category. Their algorithm does, however, contain potential weaknesses, which we explored in more depth in Section 3.5.

A second major strategy for supporting ambiguous queries is to incorporate learned user preferences or models. Pretschner and Gauch [PG99] present early work in modeling user profiles as weighted nodes in an explicit taxonomy, and explore methods for employing those taxonomies in search personalization for ambiguous queries. Their work shows modest gains in relevance are possible with re-ranking and filtering based on those profiles. Liu et al. [LYM02] study the use of general and per-user profiles constructed from category hierarchies for disambiguation of user queries.

Researchers have also considered numerous ways to evaluate the performance of search diversification and subtopic retrieval algorithms. Metrics such as search length (SL) [Coo68] and k-call [CK06], and their aggregated forms, are suited to evaluate diversification of search systems under single document assumptions. The %no metric [Voo04] measures the ability of a system to retrieve at least one

relevant result in the top ten. Other metrics, such as subtopic recall and subtopic precision [ZCL03], explicitly measure the subtopic coverage of a result set or the efficiency at which an algorithm represents the relevant subtopics. Classic ranked retrieval metrics such as NDCG, MRR, and MAP are augmented in [AGH09] to average the metrics over a probability distribution of user intent. TREC recently added a diversity track, using "intent aware" metrics such as MAP-IA [AGH09], ERR-IA [CMZ09], αNDCG [CKC08], and NRBP [CKV09].

CHAPTER 4

Keyword Generation

4.1 Introduction

With a rapidly growing movement towards online video distribution, Web search engines require effective methods for identifying relevant keywords for video content. Search engines rely on these keywords as both a means to help users locate videos of interest and to generate revenue by delivering relevant advertisements. Traditionally, television networks have monetized their content by selling time slots to advertisers, who in turn rely on estimated audiences and target demographics to determine which programs they should advertise during. Advertisements online have the potential to be more directly relevant to the video content or the interest of viewers, since ads can be selected from a large pool of advertisements individually for each viewing. The effectiveness of online ads are also easier to quantify by measuring clicks from the viewers, making the Web an attractive medium for advertisers.

Current methods for identifying the relevant keywords for a video often rely on user supplied metadata, such as the video title, summary, comments, anchor text from adjacent pages, and so on. This text is often sparse compared to the much richer video content. It is difficult to adequately identify all of the relevant keywords manually, leading to less satisfied users and missed opportunities for ad placement.

In this chapter we study the effectiveness of generating keywords using the textual *content* of a video. We focus on text sources such as production scripts and closed captioning tracks, with automatically generated speech transcripts available when neither of those are accessible. Text sources tend to be more reliable than image-based analysis in practice today, and require significantly less domain-specific knowledge or offline training.

Even with the text content for a video, several challenges remain. Identifying relevant keywords from text is non-trivial, and made more difficult when only error-filled speech transcripts are available. Methods for identifying keywords on Web pages often rely on external links and explicit structural markup or formatting [KL05, YGC06], which the text from a video lacks. Unlike documents, which generally convey information through a single medium (text), the intended user experience for a video is communicated through both visual and auditory components. Dialog is often sparse and may fail to capture this complete experience, so the relevant keywords for searchers and advertisers may not necessarily directly appear in the text for a video, particularly when only dialog-based data is available.

We address these issues using a two stage approach for identifying relevant keywords for a video. In the first step we use statistical analysis and generative models to determine a set of dominant keywords within a text source (script, closed captioning track, or speech transcript). We then discuss methods for identifying related keywords from external data sources to recapture some of the implicit information not present in the text and improve the ability to match content with advertisers. Our experiments compare how each of the individual text inputs perform as sources for keywords across a wide range of videos, including professionally produced films, news clips, and amateur videos on YouTube [You].

We compare the keywords generated from these sources for relevance to the content as well as using metrics designed to evaluate the keywords usefulness for advertising.

4.2 Overview

Our goal is to not only help a search engine deliver the most relevant content for its users, but also bridge the gap between video owners who wish to monetize their content and advertisers who want their ads displayed only when the subject matter of the video is highly relevant to the keywords they have chosen. Advertisements associated with online videos may be displayed in numerous ways, including as traditional "commercial breaks", text ads shown alongside a video, or through more engaging interactive experiences. Regardless of the means, the effectiveness of online advertising is dependent on the relevance of an advertisement to the corresponding content [WZC02].

Automatic methods for analyzing Web pages and generating relevant keywords have been commercially deployed for many years. These techniques often depend on features which are not available for text content from videos. Many videos, particularly professionally produced videos such as episodes of a television series, are only available online for a limited period of time. This makes techniques relying on collaboratively supplied data, such as tags or anchor text, less effective. Therefore the problem requires some means of unsupervised, *content-based* analysis.

Automatic methods for analyzing videos are less refined than text-based methods in practice. Image-based video analysis frequently uses supervised learning and requires significant amounts of (often domain-specific) tagged training data

to identify objects or high level concepts (e.g. [PV98, JLM03, SWG04, JNY07]). Even the most successful systems on established datasets perform poorly on real world data [ZZP08]. We note that visual analysis systems are continually evolving and improving, and may be useful in specific applications today such as logo identification or face recognition. Incorporating visual analysis with available text in a multimodal framework for selecting relevant keywords is not studied in this dissertation, but remains an area for future research.

Given the current limitations in multimedia analysis we chose to focus on textual data. Video owners may have significant textual metadata, such as scripts or closed captioning tracks. If neither of these sources are available, dialog may still be extracted from the audio track of a video using an automatic speech-to-text (STT) system, albeit often with significantly degraded accuracy. We will make use of this data to select a set of relevant keywords.

Unlike Web pages, the textual content of a video is "standalone" in nature. That is, there are no other pages with hyperlinks and anchor text providing clues about the content. There are also no explicit markup language cues to identify headings or denote emphasis in the text itself, though methods for identifying keywords on Web pages often rely on such features [YGC06]. Popular features such as term frequency are also skewed in documents such as movie scripts which, for example, often repeat phrases such as "FADE TO" to indicate the style of transition between scenes. Production scripts are written in plaintext using an *implicit* human-readable structure, and both closed captioning and speech transcripts are unstructured text.

In the first stage of processing we describe statistical and generative methods to identify the dominant keywords within the source text. Note that we use the term *keyword* to refer to text of arbitrary length, which may be individual

words or multi-word phrases. When applicable, we use a series of pre-processing and parsing steps to identify and tag elements from the implicit script structure to extract and process only the meaningful text from the source. For short or noisy (error-prone) text input, such as speech transcripts from amateur videos, we expect the performance of statistical methods to degrade. For this type of input we propose a keyword selection method based on generative topic modeling to identify underlying topics and their associated keywords. These techniques are described in Section 4.3.

The vocabulary of the extracted keywords is often limited and does not always coordinate well with the keywords searchers or advertisers have in mind. That is, while we may have a set of relevant keywords for the video, they may not overlap with the keywords of a user query or the terms advertisers intend to bid on. To address this *vocabulary impedance problem* [RCG05] we perform a second step, mining multiple data sources for related keywords. In this stage, our goal is to increase the likelihood of matching user-supplied keywords while minimizing decline in relevancy when matches do occur. This related term mining process is described in Section 4.4. In both steps, keywords are identified and ranked without consulting query logs, an inventory of ads, or advertiser supplied keywords.

4.3 Processing Source Text

In the first stage of processing, we analyze the format and complexities of video-based text sources, such as scripts, and describe methods of text analysis based on traditional statistical analysis and generative models. In this work we consider three sources of text data for a video:

- **Movie Script** - a script or screenplay is a document that outlines all of the visual, audio, behavioral, and spoken elements required to tell a story. Since film production is a highly collaborative medium, the director, cast, editors, and production crew will use various forms of the script to interpret the underlying story during the production filming process. Numerous individuals are involved in the making of a film, therefore a script must conform to specific standards and conventions that all involved parties understand and thus will use a specific format with respect to the layout, margins, notation, and other production conventions. This document is intended to structure all of the script elements used in a screenplay.

- **Closed Captioning (CC) track** - a document which contains a series of timecodes and text of the spoken dialog. Each timecode indicates when and for what duration the corresponding text appears on screen. Closed captioning tracks lack additional cues, such as visual information or indicators of the current speaker.

- **Speech-To-Text (STT)** - is a process by which audio data containing dialog or narrative content is automatically converted to a text transcription. The output typically consists of a series of words, each with an associated timecode and duration. The source audio may be of poor quality or contain non-speech sounds such as music or sound effect artifacts, which generally contribute to transcription errors. Transcription quality is typically measured by the overall word error rate (WER). A frequent goal of STT systems is to reduce the impact of a high WER, though error rates on heterogeneous content are typically quite high.

Figure 4.1 outlines the processing workflow for a complete movie script input, which includes non-speech elements such as scene headings and action descrip-

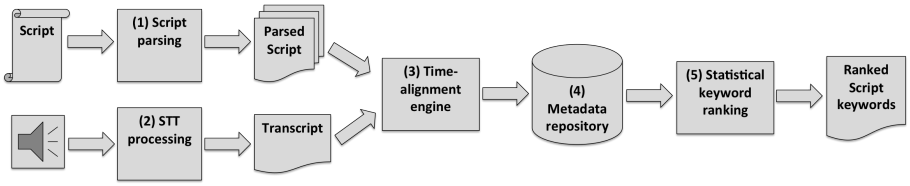

Figure 4.1: Script processing workflow

tions. We will describe each of these steps in the following sections. Note that the workflow is largely the same for closed captioning tracks and speech transcripts, which can be formatted to "look" like a screenplay. In those cases, script-specific processing steps (e.g. generating a dictionary of character names) are simply omitted.

4.3.1 Script Parsing

Television and movie scripts are frequently written in plaintext and follow a conventional "screenplay" visual format which allows human readers to easily differentiate and infer the proper semantics for different script elements, such as dialog or scene headings. For example, scene headings are typically written on a single line in all capital letters, beginning with `INT` or `EXT` to denote whether the setting is interior or exterior, and ending with an indicator of time of day such as `MORNING` or `NIGHT`. Figure 4.2 shows a brief snippet of a typical script.

Understanding the semantics of a text element is helpful when processing it. For example, character names appear frequently in a script prior to each of their lines of dialog, though we generally find them to be a poor choice for advertising keywords. We add a machine-readable hierarchical structure and semantics to each text segment of a script using a finite state machine based parser derived from conventional screenplay writing rules. This is depicted as step (1) in Figure 4.1.

```
INT. CLASSROOM - DAY

Justin sits uncomfortably at a small desk. Mr. Weiss drops
several books onto it.

                    MR. WEISS
          Here's everything you were supposed
          to read this year.

                    JUSTIN
          All of these?
```

Figure 4.2: Example script snippet

Movie script documents are converted into a structured and tagged representation where all script elements (scene headings, action descriptions, dialog lines, etc.) are systematically extracted, tagged, and recorded as objects into a specialized document object model (DOM) for subsequent processing. All objects within the DOM (e.g. entire sentences tagged by their corresponding type and script section) are then processed using both statistical methods to identify keywords of interest, and a natural language processing (NLP) engine that identifies and tags the noun items identified in each sentence. These extracted and tagged noun elements are then combined with time-alignment information and recorded into a metadata repository. We describe this alignment process next.

4.3.2 Speech-to-Text (STT) Processing

STT transcripts contain timecode information that plays an important role in associating script keywords to specific points in time in the video content. In this section of the workflow, a video or audio file that contains spoken dialog that corresponds to the dialog sections of the input script is read and processed using a Speech-to-Text engine that generates a transcription of the spoken dialog, shown as (2) in Figure 4.1. For this process, we can also perform an important optimization. Automatic speech recognition engines typically incorporate a known

vocabulary and probabilistic models of speech (often based on word N-grams). When the dialog data is available, such as from a script or closed captioning track, we construct a custom language model to bias the transcription engine towards the expected vocabulary and word sequences, which helps to increase the transcription accuracy.

4.3.3 Script and STT Transcript Alignment

At this stage, we have tagged and structured script data (without any time information) from step (1), and a noisy, relatively inaccurate STT transcript with very precise timecode information from step (2). To make better use of the keywords and concepts generated by the later processing steps, the script data must be time-aligned with the STT data. This is accomplished in step (3) by using the Levenshtein word edit distance [Lev66] algorithm to find the best word alignment between script dialog and the STT transcript. The result of this phase of processing is a time-aligned source script that can associate script action, dialog, and scene heading keywords with precise points in time within the video content. This data is stored into a metadata repository.

4.3.4 Statistical Generation of Keyword Terms

In the final processing step for a source text (script, CC track, or speech transcript), the time-coded text elements from the metadata repository are used to build a suffix word N-gram tree that is pruned by N-gram term frequency to discover the most dominant terms, based on the work of Chim and Deng [CD07]. This is shown as (5) in Figure 4.1.

Before N-gram term generation, we performed a one-time process of selecting a stopword vocabulary specific to the domain of movie scripts. Using frequency

statistics computed from a large corpus of scripts, we manually identified a set of stopwords from the most frequently occurring terms.

During N-gram term generation, the following steps are followed:

1. Corpus stopwords are removed from the source text.
2. An N-gram term tree with sequences up to length $N = 4$ is created by collecting and counting N-gram occurrences from the source text.
3. The resulting suffix tree is then pruned by traversing the tree to collect and rank the topmost M terms. In our experiments, we select and evaluate the top $M = 20$ keywords.

4.3.5 Generative Models For Noisy Data

The described statistical N-gram methods work well when keywords and phrases are repeated multiple times. While this is often the case for longer or well-formed text input, short or noisy text often results in the majority of (non-stopword) keywords only being mentioned once. With this type of input, N-gram methods are unable to decipher which keywords are most important.

To better handle short or noisy text input, we use a keyword selection method based on generative topic modeling. In this model, we assume that a video comprises a small number of hidden topics, which can be represented as keyword probabilities, and that a video's text is generated from some distribution over those topics. The highly probable keywords in those topics are likely to be most representative of the video content. We use Latent Dirichlet Allocation (LDA) [BNJ03] to learn the topics and corresponding topic-keyword probability distribution from the input text. We then combine these topics to form a ranked keyword list.

4.3.5.1 Generating Topics

To discover the underlying topics in a video, we segment the input text into sentences and perform topic modeling with LDA on those sentences. The resulting topic-keyword distribution ϕ is a KxV matrix, where K is the number of topics, V is the size of the input vocabulary, and $\phi[i][j]$ is the probability of keyword j in topic i. We form an ordered list of keywords k_i for each topic, sorted by their probability in $\vec{\phi[i]}$. This results in K ranked lists of keywords, one per topic, which must then be merged into a single list to select the top M. While simply selecting the top $\frac{M}{K}$ keywords from each topic is one option, we describe a more general solution for merging multiple ranked lists when we discuss our approach to finding related keywords. This method is described in Section 4.4.3.

4.3.6 Statistical-Generative Hybrid Method

The LDA model learns keyword probabilities for terms which are separated by whitespace. When possible, however, it is preferable to identify multi-term keywords, particularly for advertising. For example, the phrase "relational database" is more specific than either of the individual words "relational" or "database", and thus has higher value to advertisers.

To help identify these multi-token keywords in short or noisy text sources, we use a hybrid of statistical and generative techniques. We first process the source text using the N-gram method to identify any significant multi-token keywords. We then edit the source text by removing the whitespace between the terms of these multi-token phrases so they appear as a single token. This modified source text is then processed using the generative model.

4.3.7 Filtering the Keywords

We apply two filters, when possible, to remove frequently occurring words which are often not useful in the context of matching advertisements. From all input sources, keywords matching a list of English profanity are removed. We also find that main character names are often amongst the top keywords, but generally do not retrieve relevant advertisements. When given a complete script, we remove character names from the keyword list using a dictionary constructed during the parsing and tagging stage. For closed captioning and speech transcripts, however, these names are unknown and thus may still appear in the top keywords. This is more common for closed captioning than speech transcripts, however, as proper names are less likely to be correctly transcribed by the STT engine.

At this point the most dominant (possibly multi-term) keywords which occur in the source text, along with associated timecode information, have been identified and can be suggested as keywords relevant to a particular time point of a video. As Ribeiro-Neto et al. [RCG05] describe, however, the keywords chosen directly from a source and the keywords submitted by searchers or bid on by advertisers may suffer a *vocabulary impedance problem*. In the next section we describe two related term mining approaches which can provide a richer, more complete set of relevant keywords.

4.4 Discovering Related Terms

The keywords selected by analysis of the source text can provide a useful set of terms to represent the content of a video. These keywords are limited, however, to the vocabulary used by the original script authors. In the case of closed captioning and speech transcripts, they are limited further to only the words

spoken by the actors. A search engine user or advertiser may have a particular set of semantically related keywords in mind which do not necessarily overlap with any of the selected keywords. These vocabulary mismatches result in missed opportunities to connect users and advertisers with relevant content.

In this section we describe two techniques for identifying related terms to help bridge the gap between the vocabularies used in videos and keywords chosen by users and advertisers. Beginning with the ranked keywords from Section 4.3, we investigate two sources for discovering related terms: the Web corpus and Wikipedia [Wik]. Both the Web and Wikipedia are continually expanding and evolving [CG00, FMN03, AMC07], meaning new popular terminology or idioms become available as candidate related terms as their common usage increases. Likewise, it is worth noting that, as in Chapter 2, our experiments and evaluation were conducted in English but the approaches we will describe are language-independent. Both information sources used for related term mining contain vast amounts of data in other languages.

4.4.1 Mining with Web Search

Web search engines are remarkably efficient at retrieving documents relevant to an input query. The simple yet powerful notion that *semantically similar* queries will produce *textually similar* documents has been used in applications such as measuring the semantic similarity of short, possibly non-overlapping text segments [SH06]. While that type of application uses the content of Web search results as an opaque context, we are primarily interested in the terms themselves.

To find candidate related keywords for term(s) T, we first submit T as a query to a Web search engine. For each of the top k search results, we identify a set of relevant keywords for the page. We then look at the collection of search results

to rank these candidate related keywords. The high level overview is shown in Figure 4.3. In our experiments, we found that $k = 50$ generally proved to be a sufficient number.

4.4.1.1 Selecting Related Terms

Along with a title, URL, and snippet of text from each result page, the search engine used for our experiments provides a set of approximately 20 distinct keywords for each search result. These keywords are relevant to the corresponding page and are thus potentially related to the keywords which retrieved the page. In the absence of keywords explicitly provided by the search engine, social bookmarking tools such as delicious [Del] or Xmarks [Xma], related work in keyword identification for Web pages, in addition to the techniques outlined in Section 4.3.4 may be useful resource for associating keyword tags with the top search result URLs. Additionally, a similarity algorithm such as SimRank [JW02] ran over a bipartite graph constructed from query log data between queries and the clicked URLs may provide related terms. A similar technique has been used for making query suggestions [MYK08].

We model each search result as a "document" consisting of these identified keywords, shown as (2) in Figure 4.3. Each keyword is normalized through a series of standard filters such as punctuation removal, case folding, plural stemming, and stopword removal. We began with a minimal stopword list derived from frequent terms in the Reuters corpus [MRS08], but our experiments indicated that the Web corpus as a whole has a unique set of frequent terms, and our stopword filtering should be adjusted to account for this. In a sample of approximately 250,000 search result abstracts, *information, music, free, reviews, video*, and *search* all occur amongst the most frequent terms. Table 4.1 shows the top 20

1 - 4	information	school	find	music
5 - 8	news	home	free	online
9 - 12	high	reviews	world	site
13 - 16	search	time	video	page
17 - 20	photos	people	great	city

Table 4.1: Top 20 search result stopwords

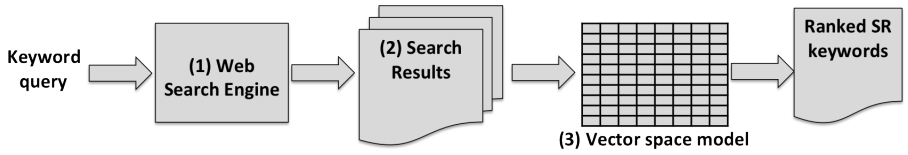

Figure 4.3: Generating related terms from search results

words by frequency, excluding numbers and words in the NLTK [BLK09] English stopwords list. Note that, after the processing steps described here, the original list of distinct keywords for a Web page may now contain duplicates.

After these filtering steps, we construct a vector space model M for this small corpus of "documents" relevant to T. Based on the popular TF-IDF [SB88] term weighting, we compute the corpus frequency (CF) and inverse-document-frequency (IDF) weight for each term in M, and rank the keywords according to their CF*IDF score. This step is shown in the workflow as (3) in Figure 4.3, producing the final list of ranked related keywords from search results.

4.4.2 Mining with Wikipedia

The second data source we analyze for related terms is Wikipedia, an extensive knowledge base with over 3.1 million English articles available at the time of this writing. Whereas in the Web corpus we focused on search results in response to

a query, with Wikipedia we direct our attention to hyperlinks. Within the text of a Wikipedia article, numerous *inter-wiki* links point to other Wikipedia pages, which allows us to model Wikipedia as a directed graph $G = \{V, E\}$.

We construct the Wikipedia graph where nodes V represent pages in the main article namespace, and edges E denote the inter-wiki links between those pages. When building this graph, two article types in the main namespace are processed specially. For ambiguous terms such as "coach", a disambiguation page in Wikipedia lists the available articles for different senses of the term. These pages serve primarily as navigational aides for users, rather than conveying any semantic relationship between terms, and we therefore exclude them in the graph. The second category of pages we process specially are redirection pages, which provide a translation for alternate or misspelled words, inconsistent capitalization, acronyms, and so on, into a canonical form. In our Wikipedia graph, an article and all of the pages which redirect to it are merged into a single logical node.

We use the link structure of the graph to both identify and rank candidate related terms. These steps are described in detail in the following sections.

4.4.2.1 Identifying Candidate Related Terms

Without clearly defined directed links between individual terms in the Web corpus, the approaches using Web search results described above depend on the assumptions that documents retrieved by the search engine are relevant to the input terms, and that other tags or keywords for those pages are potentially related. That is, they rely on co-occurrence based measures to identify which terms are most likely related.

With Wikipedia, however, we have an explicit link structure between articles which can be used as an indicator of relatedness. We require the relatedness

between two article nodes a and b to be a symmetric relationship: a is related to b if and only if b is related to a.

Translating this requirement to the Wikipedia graph is relatively straightforward. To identify candidate related terms for term T, we first locate the Wikipedia page with T as the title. Note that we could relax this requirement and search the text of Wikipedia articles to identify the top page or pages for any particular input term, albeit at a likely reduction in quality of the generated related terms. Given the node t for T, we identify any nodes in the graph which form a direct cycle with t as candidate related terms. That is, we select the subset of nodes $N \subseteq V$ such that:

$$\forall n \in N \implies \{t, n\} \in E \wedge \{n, t\} \in E$$

Figure 4.4 shows a simple example, where for term t, terms n_1 and n_2 are candidate related terms, but X and Y are not.

4.4.2.2 Ranking Candidate Terms

A given set of candidate related terms may be quite large. We now look at how to rank the candidate terms. To be a good suggestion as an advertising keyword, a term should be relatively popular. While we could measure popularity through external sources, such as query log frequency, we chose to utilize the graph structure of Wikipedia. We approximate the relative importance of terms by computing PageRank [PBM98] over the Wikipedia graph. Candidate terms are assigned a score equal to their PageRank value.

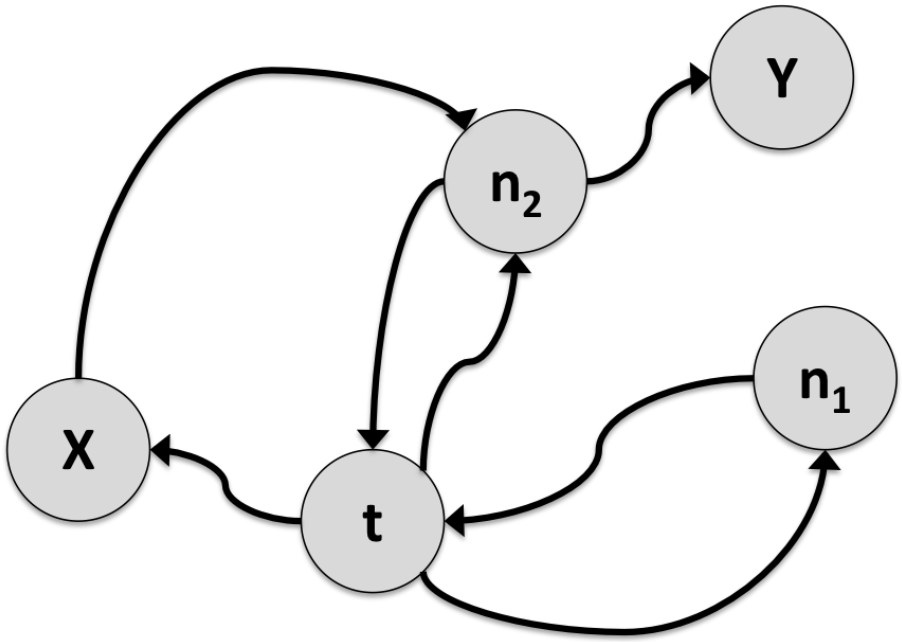

Figure 4.4: Example candidate term graph

4.4.3 Combining Ranked Lists

With two distinct sources of related terms, we now look at how to merge these ordered lists into a single ranked list. The scoring mechanism for search result keywords has no inherent range, whereas PageRank assigns a value to each node such that the score of all pages sums to one. Instead of attempting to normalize the individual scores assigned by each method to a common system, we treat each set of keywords as a ranked list, and assign each term within the list a score based on its reciprocal rank. For an ordered list of terms l, we assign a score to the term at rank i as:

$$s_l(t_i) = \frac{1}{1 + \log i} \tag{4.1}$$

Any term not existing in the list is assigned a score of 0. We may then combine the terms from any n ranked keyword lists into a single list, with a final score for

each term t as:

$$S(t) = \sum_{j=1}^{n} \alpha_j s_j(t) \tag{4.2}$$

The weight placed on list j is defined as α_j, such that:

$$\sum_{j=1}^{n} \alpha_j = 1$$

Equation 4.3 shows the combined scoring metric used for ranking terms. In our experiments, we placed equal weight ($\alpha = 0.5$) on both the search result (SR) and Wikipedia (WP) sources.

$$S(t) = \frac{\alpha}{1 + \log rank_{SR}(t)} + \frac{1 - \alpha}{1 + \log rank_{WP}(t)} \tag{4.3}$$

Tables 4.2 and 4.3 show a few examples of the suggested related terms generated by the search result and Wikipedia methods described above, as well as the combined ranking computed with Equation 4.3.

4.5 Evaluation

In this section we present the results of evaluations designed to assess the keywords selected by our methods for each text source. As there are no publicly available data sets suitable for such a task, and we do not have access to an ad corpus, we conducted a user survey to evaluate the chosen keywords and design a series of metrics to quantify their effectiveness. We evaluate using text from production scripts, closed captioning tracks, and speech-to-text transcripts across a range of videos including 12 full length films, 3 clips from news and educational content, and 5 popular (over 100,000 views) amateur clips from YouTube.

Search results	**Wikipedia**	**Combined**
digital camera	photography	digital camera
lens	pornography	photography
canon	visual arts	canon
nikon	photograph	nikon
zoom	digital camera	pornography
film camera	photojournalism	lens
digital slr	photographic film	digital photography
megapixels	aperture	photograph
digital photography	canon	aperture
compact	photographic lens	shutter speed
camcorder	aerial photography	visual arts
slr camera	holography	exposure
lense	single-lens reflex camera	viewfinder
digital slr camera	focal length	movie camera
olympus	nikon	camera phone

Table 4.2: Example related terms for keyword “camera”

Search results	**Wikipedia**	**Combined**
product	internet	internet
marketing	newspaper	product
advertiser	video game	marketing
business	american football	newspaper
campaign	magazine	advertiser
advertising agency	world wide web	magazine
internet	marketing	advertising agency
consumer	mtv	public relations
job	blog	google
newspaper	public broadcasting service	billboard
agency	mass media	video game
public relations	google	publicity
company	brand	product placement
service	broadcasting	graphic design
budget	music video	promotion

Table 4.3: Example related terms for keyword “advertising”

4.5.1 Evaluation Design

We identified the top 20 keywords from each available text source using both the statistical and hybrid approaches described in Section 4.3. For each of these keywords we use the related term mining techniques of Section 4.4 to identify the top 10 related terms. These keywords were then evaluated with a user survey. For the topic modeling phase of the hybrid technique, we set the number of topics $K = 5$ with the LDA parameters $\alpha = 0.3$ and $\beta = 0.1$.

Users were shown a video clip, typically around 3 minutes in length, and a set of keywords. To keep the size of the keyword set manageable, we show 5 of the top 20 keywords for each method from each available text source, and 1 of the top 10 related terms for each of those keywords, all chosen and ordered at random. Users were asked to make a binary assessment on the relevance of each displayed keyword. For the news and educational videos and the amateur clips available from YouTube, users were shown the complete video. For full length films, users were shown the theatrical trailer and asked to make judgements based on that trailer and their prior knowledge of the movie. The survey was announced to volunteers through social networking sites and mailing lists within the UCLA Computer Science Department. At least 23 people participated in the survey (providing personally identifiable information was optional, and so the exact count is unknown), with a minimum of 9 and an average of 13 users evaluating each video.

4.5.2 Evaluation Metrics

We evaluate the keywords generated by our methods using four metrics. The average relevancy of the keywords displayed to users we call the *precision*. Multiple users viewing the same set of keywords may not completely agree on which

keywords are relevant. We therefore compute the *potential* of a source, which measures the fraction of the keywords judged relevant by *at least one* user. More formally, we define the precision and potential of text source S as:

$$\text{Precision}(S) = \frac{1}{i} \sum_{i} \frac{|K_i(S) \cap R_i|}{|K_i(S)|} \tag{4.4}$$

$$\text{Potential}(S) = \frac{|R(S)|}{|K(S)|} \tag{4.5}$$

R_i is the set of keywords judged relevant in evaluation i and $K_i(S)$ are the keywords displayed to the user for evaluation i which come from source S. $K(S)$ are the keywords from source S displayed in at least one evaluation, and $R(S)$ are the keywords from source S judged relevant by at least one user, defined as:

$$R(S) = \bigcup_{i} K_i(S) \cap R_i$$

$$K(S) = \bigcup_{i} K_i(S)$$

The other metrics we define are *appeal* and *popularity*, which serve as indicators of how pertinent the keywords are to advertisers. Appeal estimates the likelihood that a keyword deemed relevant to the content will also be meaningful to an advertiser. Popularity measures the average number of advertisers interested in a relevant keyword. We define the appeal and popularity of a source S as:

$$\text{Appeal}(S) = \frac{|R(S) \cap A^*|}{|R(S)|} \tag{4.6}$$

$$\text{Popularity}(S) = \frac{1}{|R(S)|} \sum_{k \in R(S)} A_k^* \tag{4.7}$$

A^* is the set of all keywords advertisers have bid on, and A_k^* is the number of advertisers bidding for keyword k. Since we do not have an inventory of ads available to exactly measure A^* or A_k^*, we estimate them using a Web search engine. A^* is approximated as the set of all keywords which retrieve at least one advertisement when issued as a search query, and A_k^* is the number of ads returned for query k. Although most commercial search engines limit $0 \leq A_k^* \leq 8$, we are primarily concerned with relative performance across text sources. Note that, because the appeal and popularity of a keyword are meaningless if it is not relevant to the content, we compute these metric values for the set of keywords identified as relevant by at least one user. For all metrics, higher values indicate better performance.

4.5.3 Overview of Results

Before we delve into the detailed results, we briefly highlight several interesting trends that we observed from our experiments.

1. As we conjectured in Section 4.3.5, statistical keyword selection produces higher precision for longer text inputs, while the generative hybrid method performs significantly better on shorter, user generated content. In particular, the generative hybrid method model showed meaningful improvement when only speech transcripts are available for a short video clip.

2. Closed captioning has the highest overall precision, though speech transcripts are nearly as effective and produced relevant keywords for video categories where the background noise is reasonably contained and the STT language models are properly trained, such as news and educational content.

3. Regardless of the text source or video type, related terms consistently appear more profitable for advertising. The precision of the related terms, however, is not as high as the terms directly from the source.

We expand on these observations with more detailed results in the following sections.

4.5.4 Precision and Potential of Text Sources

Table 4.4 shows the precision and potential for all three text sources and the two keyword selection methods. For all tables (except Table 4.6), significance tests were performed, and cells in bold indicate a statistically significant difference in performance ($p < 0.05$) between the two methods. For example, in Table 4.4, the precision of the statistical method on closed captioning tracks was higher than the hybrid method with $p = 0.037$.

As we expected, for "well formed" text such as scripts and closed captioning tracks, the statistical method generally achieves higher precision, while the LDA-based hybrid method shows slightly better performance on the noisier speech transcripts, though the difference is not large enough to be statistically significant. Interestingly, we also see that the closed captioning data actually outperforms the full scripts. This may indicate that viewers more closely associate dialog with the main points or themes of a video than the additional props, scenery, and actions described in a complete script.

To investigate further whether the observed improvement of the hybrid method on STT input is meaningful for certain classes of video, we take a closer look at the performance for speech transcripts across three different video types in Table 4.5. Here we see that for the longer, professionally produced films, the sta-

Source	Precision		Potential	
	Statistical	Hybrid	Statistical	Hybrid
Script	0.389	0.353	0.662	0.635
CC	**0.443**	0.397	0.758	0.705
STT	0.291	0.307	0.467	0.514

Table 4.4: Precision and potential

Video Type	Precision		Potential	
	Statistical	Hybrid	Statistical	Hybrid
Studio Films	0.268	0.252	0.479	0.480
News/Educational	0.442	0.473	0.548	0.717
User Generated	0.268	**0.368**	0.390	0.473

Table 4.5: Precision and potential for speech transcripts

tistical method achieves marginally higher precision even on speech transcripts. The hybrid method performs significantly better on the shorter (3-4 minute) user generated clips, which supports our earlier intuition that statistical methods alone would likely have insufficient data to find the best keywords in such cases. We also note that news and educational content, on which the speech-to-text engine is expected to be most accurate, achieves the highest overall precision and potential.

Hauptmann's work indicates that speech-to-text word error rates under 0.4 result in retrieval performance comperable to a perfect transcript [Hau05]. At the 0.4 threshold, relative retrieval precision is approximately 80%. We compute the average word error rate for studio films and news/educational videos (using the default "general" language models for our STT engine), and compare the relative precision of STT with respect to closed captioning for the statistical and hybrid

Video Type	WER	Statistical	Hybrid
Studio Films	0.857	0.723	0.690
News/Educational	0.406	0.731	0.961

Table 4.6: Relative precision and word error rate (WER)

methods, shown in Table 4.6. User generated videos are not included because no "correct" transcripts are available for the content. As expected, the average word error rates for news and educational videos are substantially lower, though still around 0.4. For this type of content, the relative precision of STT is 96% of the closed captioning. For the higher word error rate of films we can still achieve over 70% average relative precision. These results further support use of the statistical selection methods on longer text inputs and the generative methods on shorter text, and suggest that speech transcripts alone may be sufficient to find meaningful keywords for videos such as news broadcasts.

4.5.5 Precision and Potential of Related Terms

We next look at the precision and potential of the related terms. Table 4.7 shows the precision and potential scores for the top 10 related terms from both the statistical (S-Related) and hybrid (H-Related) methods. These results are mostly consistent with Table 4.4, with the most precise input source (closed captioning) producing the most relevant related keywords.

For each method and source, the precision and potential of the source keywords are higher than the related terms. In our experiments we randomly selected from the top $N = 10$ related terms for each source keyword. We now investigate how the average precision of the related terms is affected as we vary this range for $1 \leq N \leq 10$. Figure 4.5 plots the precision of the related keywords for each

Source	Precision		Potential	
	S-Related	H-Related	S-Related	H-Related
Script	**0.254**	0.215	0.253	0.222
CC	**0.260**	0.221	0.262	0.221
STT	0.208	0.186	0.200	0.191

Table 4.7: Precision and potential of related terms

text source using the statistical selection method. For closed captioning, the top 2 related terms give the highest precision, which is lower than the precision of the source terms but significantly higher ($p = 0.003$) than choosing from the top 10. Both script and speech transcript inputs show an increase in precision when selecting from the top 3-6 terms. While the precision is again lower than the source keywords, there is noticeable improvement between selecting from the top $N = 6$ and $N = 10$ for both script ($p = 0.06$) and STT ($p = 0.03$) input. This result suggests that, for our methods, the number of related terms to consider to achieve the maximum overall precision depends on the input text type, with higher precision input like closed captioning achieving its best precision with a smaller number related terms than scripts or speech transcripts. Results for the hybrid selection method exhibit similar behavior.

Another factor to consider when evaluating the precision of the related keywords is the relevancy of the source term being expanded. An irrelevant source term is less likely to result in relevant related keywords. Figure 4.6 shows the average precision of the top $N = 1, 3, 5$ and 10 related terms for the source keywords selected by the statistical method and identified as relevant by at least one user. The graph shows that the precision of related terms is higher, in most cases by a significant margin (columns marked with an asterisk), when starting from relevant source terms. The precision is relatively constant for the top several

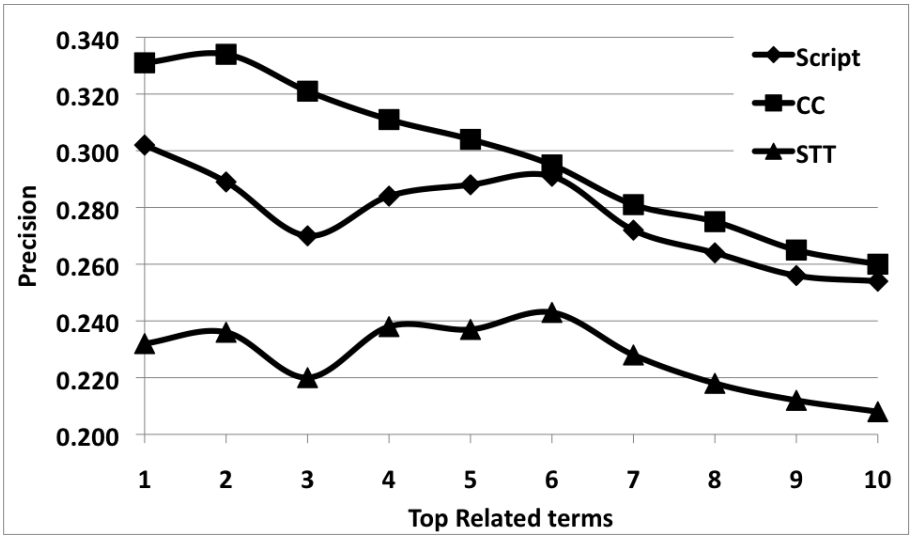

Figure 4.5: Precision of related terms

related terms, again suggesting $1 \leq N \leq 5$ is a good choice for how many related terms to select without decreasing relevancy.

4.5.6 Appeal and Popularity

We now turn to metrics for measuring the utility of the keywords to advertisers. Table 4.8 shows the appeal, or fraction of the *relevant* keywords which return at least one advertisement, for each method and its corresponding related terms. Approximately 80% of the related keywords return at least one advertisement, regardless of the text source, while the source keywords vary from 54% to 73%. The data shows the related terms from CC and STT keywords are significantly more likely to be bid on by advertisers than the source keywords themselves.

The second measure of utility for advertisers is popularity, where we evaluate the average number of ads returned for each relevant keyword. The maximum achievable score for popularity is 8, as that is the most advertisements returned for a single query by the search engine. The popularity of the related keywords,

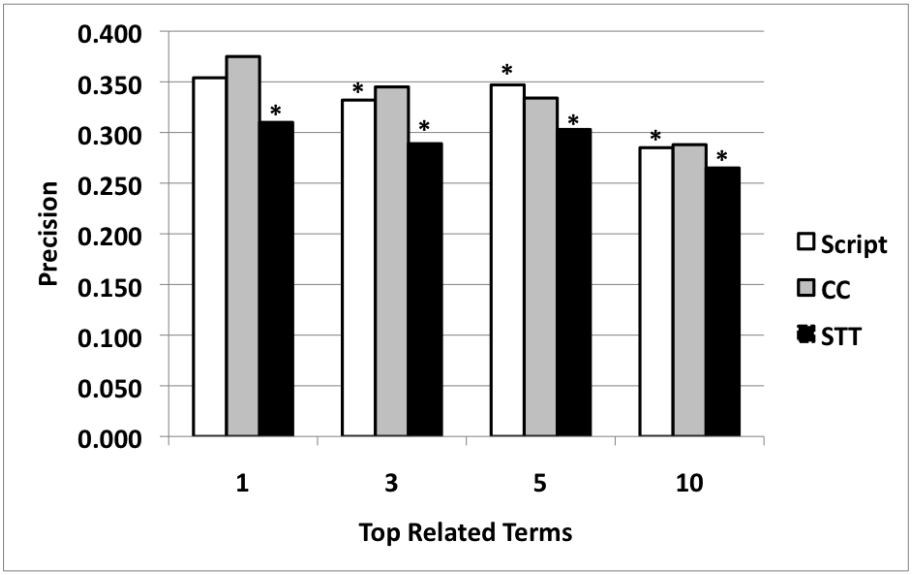

Figure 4.6: Precision of related terms from relevant source terms

Source	Statistical	S-Related	Hybrid	H-Related
Script	0.726	0.788	0.607	0.792
CC	0.578	**0.785**	0.543	**0.796**
STT	0.681	**0.827**	0.594	**0.820**

Table 4.8: Appeal of keywords by source

Source	Statistical	S-Related	Hybrid	H-Related
Script	3.59	3.96	3.00	**4.18**
CC	2.11	**3.81**	2.00	**3.77**
STT	2.54	**4.39**	2.56	**4.30**

Table 4.9: Popularity of keywords by source

Source	Statistical	S-Related	Hybrid	H-Related
Studio Films	2.97	**4.35**	2.67	**4.39**
News/Educational	1.69	**4.11**	2.21	**3.50**
User Generated	1.89	**4.83**	2.63	**4.75**

Table 4.10: Popularity for speech transcripts

shown in Table 4.9, is notably higher than the source keywords in all cases except for script input and the statistical method. The significantly higher popularity of related keywords again suggests they would be more beneficial for advertising.

For both appeal and popularity we notice that, while closed captioning was generally considered the most precise source of keywords, it also produces the least meaningful keywords for advertisers. This may be a result of character names appearing in the closed captioning keywords, which we noted earlier are filtered out from script input text and are less likely to retrieve relevant ads.

We look closer at the popularity of keywords for speech transcripts. Table 4.10 compares the popularity for source and related keywords for various video types. In all cases, the related keywords have higher popularity than the source keywords by a statistically significant margin. It also shows that news and educational content contains less popular keywords for advertisers.

4.5.7 Precision-Popularity Tradeoffs

The results above demonstrate that, when relevant, related keywords are significantly more attractive to advertisers than source keywords. The overall precision of the related terms, however, is lower than source terms. We explore the tradeoff between keyword relevance and popularity by computing a precision-weighted popularity metric:

$$PWP(S) = \frac{\sum_{k \in K(S)} A_k^* \; P(S,k)}{|K(S)|} \tag{4.8}$$

Where $P(S,k)$ is the average precision of keyword k from source S, defined as:

$$P(S,k) = \frac{\sum_i |\{k\} \cap R_i(S)|}{\sum_i |\{k\} \cap K_i(S)|}$$

Table 4.11 shows the precision-weighted popularity for the statistical method for each text source using the top 5 related keywords from each source keyword. The results suggest that for script input, the minor improvement in popularity of related keywords (shown in Table 4.9) may not offset the decrease in precision. For speech transcript input, however, there appears to be some benefit from related terms.

We examine STT input further in Table 4.12, which shows that overall, even with the drop in precision, related keywords may be beneficial to advertisers for news and user generated videos when only speech transcripts are available. Although the related keywords for studio film speech transcripts have higher popularity than source keywords (Table 4.10), the relative increase is noticeably lower than for CC or STT, and the resulting precision-weighted popularity does not offer improvement.

Source	Statistical	S-Related
Script	**1.358**	0.908
CC	0.964	0.955
STT	0.661	0.842

Table 4.11: Popularity weighted by precision

Source	Statistical	S-Related
Studio Films	0.726	0.663
News/Educational	0.546	**1.278**
User Generated	0.563	**1.164**

Table 4.12: Popularity weighted by precision for speech transcripts

4.6 Conclusion

In this chapter we have explored the suitability of a range of text sources for generating keywords for video content. Our experiments have demonstrated that statistical keyword selection methods are effective when a sufficient amount of text data is available, while generative methods appear preferable when data is short or error prone, as is often the case with automatic speech recognition and user generated clips on sites such as YouTube.

We have also shown that related term mining techniques can substantially improve the likelihood of matching relevant and more marketable advertiser keywords. For Web search engines, speech transcripts are the only data source guaranteed to be available, and our results suggest that expanding the source keywords with 5-6 related keywords can improve advertising effectiveness, even with the decrease in average precision. As a result of investigation into related term mining, we also described a relatively simple but effective approach to merg-

ing multiple ranked lists.

While the results suggest that closed captioning tracks provide the most relevant keywords, another explanation may hold. Namely, elements of the dialog from a video may be easier for viewers to notice and remember than props, scenery, or on-screen actions of the actors involved. This has potential implications on how high quality image recognition can best be applied to video indexing and advertising in the future, as it suggests users are perhaps less likely to consider keywords associated with individual objects from a scene as relevant. It also explains why news and educational videos score highest on precision, as this class of video typically conveys more information through dialog than the others.

Although only studied briefly in this work, clearly a more intricate tradeoff between precision and popularity can be played using a combination of source and related keywords. One possible solution might be to primarily use keywords from the source text and, for those keywords which are less popular with advertisers, supplement them with additional related keywords.

4.7 Related Work

Sponsored search, or advertising displayed alongside the search results of a user-supplied keyword query, typically involves a complex combination of advertisers bidding on keywords, review of advertisements for relevance, and an auction process to determine ordering or placement of ads alongside search results. See [Ber08] for an overview of sponsored search.

In display or content-match advertising, however, explicit keywords for the content are not provided. In online advertising it is important to display ads relevant to a page's content [WZC02]. Without user-supplied keywords, researchers

have investigated numerous keyword identification techniques and approaches to match advertisements with the content of Web pages. Ontologies or taxonomies have been used in combination with feature identification for semantic approaches to matching advertisements with content [BFJ07, CXY08]. Ontologies are often domain-specific and tedious to construct, and the individual text elements from scripts or dialog are often terse, making the use of classification techniques or ontologies more error prone.

Researchers have studied several features of documents and query logs, such as term and corpus frequencies, textual characteristics (e.g. capitalization), the content of neighboring pages, and structural cues for identifying keywords [FPW99, Tur03, KL05, YGC06]. Ribeiro-Neto et al. [RCG05] propose strategies for matching the text of a Web page with text-based advertisements in a known ad inventory. They address the *vocabulary impedance problem* by representing a page with concepts from its nearest (most similar) neighbors. Ravi et al. [RBG10] propose a two phase generative model for identifying relevant advertising keywords for a given Web page. They use a popular machine translation method to learn a probabilistic set of keyword mappings from a training corpus of Web pages associated with ads and advertiser chosen keywords. Term weights are assigned based on HTML features. A bigram language model trained on queries in a search query log is used to help rank the generated candidate keywords. Finding related but "less obvious" (and therefore less expensive) keywords from an advertisers point of view [JM06, AH07] has been addressed as well.

Our work differs from these problems in several ways. We investgiate plain-text sources which lack explicit structural cues such as HTML markup. The text occurrs in isolation, rather than in a hyperlinked environment like the Web. We therefore must resort to statistical methods for ranking and selecting keywords

from the source text. Several of the above techniques also require tagged training data, language and domain-specific ontologies, or pre-constructed pools of available advertisements. Our methods are language-independent and unsupervised, not requiring any training data.

Related term identification is a well researched problem in the information retrieval domain, where tasks such as query rewriting or expansion are widely studied. Voorhees [Voo94] described the use of lexical relationships contained in WordNet for query expansion. Buckley et al. [BSA94] noted that related terms will typically co-occur non-randomly in documents relevant to a query. Sahami and Heilman [SH06] use a similar notion to compute the semantic similarity of short text snippets using Web search results as an opaque context. Jones et al. [JRM06] described techniques for generating query reformulations from query logs. Graphical models for term expansion have also been studied using random walks and multiple semantic links [LZ01, CC05]. Our related term mining approaches follow along these lines by using Web search results to discover related co-occurring terms. We also use the implicit semantic relationships captured in the hyperlinked structure of Wikipedia to identify related terms.

Hauptmann summarizes many "lessons learned" regarding speech recognition accuracy and the effects of word error rate on information retrieval precision [Hau95, Hau05]. In particular their research shows that the best systems achieve word error rates around 0.15 under ideal conditions (such as in-studio anchors for broadcast news), and that retrieval performance degrades relatively gracefully with respect to perfect text transcripts until word error rates approach 0.40.

Keyword identification for multimedia often utilizes, in part, attributes extracted from images as part of a larger feature space for machine learning. Veliv-

elli and Huang [VH06] predict tags for videos based on image features and speech transcripts. Using a collection of speech transcripts, they perform a PLSI-based clustering to form k topic themes. Each cluster is used to generate a unigram language model θ_i, and a scene is assumed to be a mixture of these models and an underlying base model. Tags are then predicted using a combination of shot features and keyword co-occurrence based on a constructed training set.

The PageRank-inspired TagRank algorithm for propagating the weights of tags to neighboring YouTube videos is described in [SSS09], where a video's neighbors are determined using image-based overlap and near duplicate detection techniques. Using similar documents to find related keywords has also been explored in the video domain. Moxley et al. [MMH08] mine keywords for a video using speech transcripts. They reduce the errors induced by automatic speech recognition inaccuracies by expanding their dataset to include "similar" transcripts. The score assigned to each term is proportional to its frequency in the collection of transcripts, where the contribution to term t_i from transcript d is based on the "similarity" between d and the original video. They experiment with several measurements of similarity, using both text and image features. In our work we focus on text features and consider each video individually, rather than assuming a large corpus to search for similar content. We chose to avoid image-based features at this time, as they can require large training sets and are susceptible to noise [ZZP08].

CHAPTER 5

Conclusions and Future Work

The Web has quickly become a crucial information source for people throughout the world. Users rely on Web search engines daily to help find answers to their questions. Their input to the search engine typically consists of very short keyword phrases from which it is often difficult or impossible to capture the user's complete intent. At the same time, the search engine is constantly discovering and indexing new content, and determining the relevant keywords from this content can be challenging.

In this dissertation we have investigated several problems Web search engines face when encountering underspecified queries and content. First, we studied how a search engine can determine which queries or keywords imply additional relevant context. Next, we addressed how a search engine can diversify the results for an ambiguous query to improve user satisfaction. Finally, we proposed and evaluated methods for identifying keywords from the available textual content for videos, which is often imprecise or error prone.

In Chapter 2, we studied identification of geo-localizable queries. We observed that approximately 15% of the queries submitted to a search engine are implicitly geo-localizable, and proposed a framework for automatically identifying such queries. In our framework, we first identify a set of candidate localizable queries by tagging and removing portions of a query which match against a set of known locations (e.g. city and state names). We proposed several features measurable

from data in query logs, such as localization ratio, distribution of location occurrences, and clickthrough rates. We computed these feature scores for each candidate query, which are used by supervised learning algorithms to determine if the candidates are localizable.

Through cross validation experiments we found that individual classifiers are capable of over 80% precision for positive (localizable) classification. The errors made by the top individual classifiers were often non-overlapping, and we propose a simple majority voting scheme with multiple distinct classifiers which achieves up to 94% precision for positive classification. Our approach also accounts for two important search engine requirements, language independence and scalability, by keeping the individual steps simple and highly distributable.

In Chapter 3, we addressed the challenges of uncertainty in user intent with ambiguous informational queries. We proposed a model for user satisfaction which is well suited for the requirements of informational queries, namely, by accounting for users who may require more than one relevant document. We defined three probability distributions estimating (1) the number of relevant pages the user is expected to require, (2) the user's intent in each subtopic, and (3) how well a page satisfies the user's need for each subtopic, and described how they can be approximated from data available to a Web search engine. These distributions are used by the *Diversity-IQ* algorithm to select a set of pages for an ambiguous query. Experiments show the pages selected by *Diversity-IQ* can greatly improve the average user satisfaction for ambiguous informational queries.

In Chapter 4, we investigated methods for generating relevant keywords from a variety of text sources for video content. We evaluated both statistical selection and generative modeling techniques for identifying relevant keywords from source text. Our findings suggest that statistical methods are more appropriate for long

or well formed text input, while generative modeling performs better when the only available text is short and error prone, as is often the case with speech transcripts from user generated content.

We also show that, while the terms obtained directly from the original text source are generally the most relevant, terms *related to* the original source keywords are often more suitable for advertising. When factored with the corresponding decline in precision, however, the value of including related terms depends on the noisiness of the source. For the more complete script and closed captioning text inputs, the popularity of source terms is sufficient that adding related terms does not appear to be beneficial. Including related terms for speech transcript input, however, appears to significantly improve the overall effectiveness for advertising, particularly for news clips and user generated content.

Web search is a continually evolving field, and finding solutions to the problems outlined in this dissertation can lead to improved user satisfaction and increased revenue for a Web search engine and its advertising partners. Our hope is that the work presented in this dissertation highlights some of the challenges and contributes ideas and insight towards solving a few of the many difficult problems faced every day by Web search engines.

5.1 Future Work

In Chapter 2 we studied automatic identification of geo-localizable queries using features computed from query logs and supervised learning. The features discussed in Section 2.5 were selected based on the static nature of the available query log data. Some useful data, such as a user's IP address, was not available in the log. Determining if the user issued the query from a mobile device may

also prove a valuable feature for classifying geo-localizable queries. More complex features, such as observing query reformulations which add an explicit location context, might improve precision. With a "live" system, we may imagine additional features and design relevance experiments to collect more dynamic features for use in classification. For example, incorporating a mix of localized and non-localized results for a user query and measuring user activity (e.g. clickthoughs) may be used to evaluate user preferences, and act as a feedback loop to future iterations of the classification algorithm to further improve precision.

We focused on determining whether a particular query is localizable or not. Once these queries are identified, a next logical step is to evaluate techniques for integrating the classifier into an information retrieval system. Once a decision to localize has been made, the search engine must determine the proper degree of localization. For example, should the query be localized to the state or city level? Our tagging process maintains information about the specific locations which occur with each query, making valuable data for this task readily available.

In Chapter 3 we studied search diversification and presented the *Diversity-IQ* algorithm. One limitation of our algorithm is the minor performance penalties, with respect to other diversification techniques, on metrics designed under single relevant document assumptions. This issue can be addressed with a more detailed model for the page requirements distribution ($\Pr(J|U)$). Using a single distribution for page requirements in our experiments is a simplification, and a query-dependent $\Pr(J = j|U, q)$ or query-class dependent $\Pr(J = j|U, C(q))$ distribution may help improve the model. For example, for navigational queries, we may want to set $\Pr(J = 1|U, C(q) = nav) \approx 1.0$.

Our work focused primarily on diversifying amongst the high level subtopics. For future work, diversification could be extended in a hierarchical fashion, look-

ing both across and within subtopics to produce a document set which covers the broader subtopic categories while also considering the range of information available within a single category. Hierarchical topic modeling such as the work presented by Blei et al. [BGJ03] shows potential for extensions in that direction. Content-based approaches [CG98, CK06] or risk-minimization [ZWT09, WZ09] for sub-diversification may also lead to interesting results.

In Chapter 4 we studied techniques for generating relevant keywords from the textual content of videos. Our keyword selection and related term mining approaches do not consider submitted user queries or an advertising corpus. It is possible to bias the keyword selection towards more popular keywords by incorporating such data, as is suggested in [RBG10], which would likely increase scores on appeal and popularity. Doing so, however, may also reduce precision.

We also limited related term mining to relatively simple methods, though more elaborate approaches might lead to improved overall accuracy. In particular, while we excluded ambiguous terms from the Wikipedia graph, word-sense ambiguity problems may still occur with Web search results. As we noted in Chapter 3, search results are often dominated by a single interpretation of such queries, and if the most "popular" sense on the Web does not match the correct one for a given source keyword, the identified related terms will be inaccurate. Determining the best context or coordinating terms from the surrounding, often noisy text (or possibly from multimodal inputs such as audio or image analysis), incorporating concept ontologies, as well as finding an optimum point between precision and popularity of keywords are some of the interesting areas for further research in this area.

References

[AGH09] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. "Diversifying search results." In *WSDM '09: Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pp. 5–14, 2009.

[AH07] Vibhanshu Abhishek and Kartik Hosanagar. "Keyword generation for search engine advertising using semantic similarity between terms." In *ICEC '07: Proceedings of the 9th International Conference on Electronic Commerce*, pp. 89–94, 2007.

[Ale] Alexa. http://www.alexa.com/.

[AMC07] Rodrigo Almeida, Barzan Mozafari, and Junghoo Cho. "On the Evolution of Wikipedia." In *ICWSM'07: International Conference on Weblogs and Social Media*, 2007.

[Ber08] Tracy Mullen Bernard J. Jansen. "Sponsored search: an overview of the concept, history, and technology." In *International Journal of Electronic Business*, volume 6, pp. 114–131, 2008.

[BFJ07] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. "A semantic approach to contextual advertising." In *SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 559–566, 2007.

[BGJ03] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. "Hierarchical Topic Models and the Nested Chinese Restaurant Process." In *Advances in Neural Information Processing Systems*, 2003.

[Bin] Bing. http://www.bing.com/.

[Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, November 1995.

[BLK09] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[BNJ03] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. "Latent dirichlet allocation." *Journal of Machine Learning Research*, **3**:2003, 2003.

[Bro02] Andrei Broder. "A taxonomy of web search." *SIGIR Forum*, **36**(2):3–10, 2002.

[BSA94] Chris Buckley, Gerard Salton, James Allan, and Amit Singhal. "Automatic Query Expansion Using SMART." In *Text REtrieval Conference*, 1994.

[Bur] U.S. Census Bureau. http://www.census.gov/.

[Bur98] Christopher J. C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition." *Data Mining and Knowledge Discovery*, **2**(2):121–167, 1998.

[CC05] Kevyn Collins-Thompson and Jamie Callan. "Query expansion using random walk models." In *CIKM '05: Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pp. 704–711, 2005.

[CD07] Hung Chim and Xiaotie Deng. "A new suffix tree similarity measure for document clustering." In *WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pp. 121–130, 2007.

[CG98] Jaime Carbonell and Jade Goldstein. "The use of MMR, diversity-based reranking for reordering documents and producing summaries." In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998.

[CG00] Junghoo Cho and Hector Garcia-Molina. "The Evolution of the Web and Implications for an Incremental Crawler." In *VLDB '00: Proceedings of the 26th International Conference on Very Large Databases*, pp. 200–209, 2000.

[CK06] Harr Chen and David R. Karger. "Less is more: probabilistic models for retrieving fewer relevant documents." In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 429–436, 2006.

[CKC08] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. "Novelty and diversity in information retrieval evaluation." In *SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 659–666, 2008.

[CKV09] Charles L. Clarke, Maheedhar Kolla, and Olga Vechtomova. "An Effectiveness Measure for Ambiguous and Underspecified Queries." In *ICTIR '09: Proceedings of the 2nd International Conference on Theory of Information Retrieval*, pp. 188–199, 2009.

[CMZ09] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. "Expected reciprocal rank for graded relevance." In *CIKM '09: Proceeding of the 18th ACM Conference on Information and Knowledge Management*, pp. 621–630, 2009.

[com] comScore. http://www.comscore.com/Press_Events/Press_Releases/2009/9/Google_Sites_Surpasses_10_Billion_Video_Views_in_August/.

[Coo68] William S. Cooper. "Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems.", 1968.

[CXY08] Yifan Chen, Gui-Rong Xue, and Yong Yu. "Advertising keyword suggestion based on concept hierarchy." In *WSDM '08: Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pp. 251–260, 2008.

[Del] Delicious. http://delicious.com/.

[Fel98] Christiane Fellbaum. *WordNet: An Electronical Lexical Database*. The MIT Press, Cambridge, MA, 1998.

[FMN03] Dennis Fetterly, Mark Manasse, Marc Najork, and Janet Wiener. "A large-scale study of the evolution of web pages." In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pp. 669–678, 2003.

[FPW99] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. "Domain-Specific Keyphrase Extraction." In *IJCAI '99: Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pp. 668–673, 1999.

[FS96] Yoav Freund and Robert E. Schapire. "Experiments with a New Boosting Algorithm." In *International Conference on Machine Learning*, pp. 148–156, 1996.

[GGP04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. "Combating web spam with trustrank." In *VLDB '04: Proceedings of the 30th International Conference on Very Large Databases*, pp. 576–587, 2004.

[GHL03] Luis Gravano, Vasileios Hatzivassiloglou, and Richard Lichtenstein. "Categorizing web queries according to geographical locality." In *CIKM '03: Proceedings of the 12th International Conference on Information and Knowledge Management*, pp. 325–333, 2003.

[Goo] Google. http://www.google.com/.

[GS04] Thomas L. Griffiths and Mark Steyvers. "Finding scientific topics." *Proceedings of the National Academy of Sciences of the United States of America*, **101**:5228–5235, April 2004.

[Hau95] Alexander Hauptmann. "Speech recognition in the Informedia Digital Video Library: uses and limitations." In *TAI '95: Proceedings of the 7th International Conference on Tools with Artificial Intelligence*, p. 288, 1995.

[Hau05] Alexander Hauptmann. "Lessons for the Future from a Decade of Informedia Video Analysis Research." In *CIVR '05: Proceedings of the 4th International Conference on Image and Video Retrieval*, pp. 1–10, 2005.

[HCO03] Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. "Relevant term suggestion in interactive web search based on contextual information in query session logs." *Journal of the American Society for Information Science and Technology*, **54**(7):638–649, 2003.

[JGP05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. "Accurately interpreting clickthrough data as implicit feedback." In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 154–161, 2005.

[JL95] George H. John and Pat Langley. "Estimating Continuous Distributions in Bayesian Classifiers." In *UAI '95: Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, 1995.

[JLM03] J. Jeon, V. Lavrenko, and R. Manmatha. "Automatic image annotation and retrieval using cross-media relevance models." In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 119–126, New York, NY, USA, 2003. ACM.

[JM06] Amruta Joshi and Rajeev Motwani. "Keyword Generation for Search Engine Advertising." In *ICDMW '06: 6th IEEE International Conference on Data Mining - Workshops*, pp. 490–496, 2006.

[JNY07] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. "Towards optimal bag-of-features for object categorization and semantic video retrieval." In *CIVR '07: Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, pp. 494–501, 2007.

[Joa02] Thorsten Joachims. "Optimizing search engines using clickthrough data." In *KDD '02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 133–142, 2002.

[JRM06] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. "Generating query substitutions." In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pp. 387–396, 2006.

[JS06] Bernard J. Jansen and Amanda Spink. "How are we searching the World Wide Web? A comparison of nine search engine transaction logs." *Information Processing and Management*, **42**(1):248–263, 2006.

[JSS00] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. "Real life, real users, and real needs: a study and analysis of user queries on the web." *Information Processing and Management*, **36**(2):207–227, 2000.

[JW02] Glen Jeh and Jennifer Widom. "SimRank: a measure of structural-context similarity." In *KDD '02: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 538–543, 2002.

[JW03] Glen Jeh and Jennifer Widom. "Scaling personalized web search." In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pp. 271–279, 2003.

[JZR08] Rosie Jones, Wei V. Zhang, Benjamin Rey, Pradhuman Jhala, and Eugene Stipp. "Geographic intention and modification in web search." *International Journal of Geographical Information Science*, **22**(3):229–246, 2008.

[KCM06] Reiner Kraft, Chi Chao Chang, Farzin Maghoul, and Ravi Kumar. "Searching with context." In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pp. 477–486, 2006.

[KHZ00] Yu-Hwan Kim, Shang-Yoon Hahn, and Byoung-Tak Zhang. "Text filtering by boosting naive Bayes classifiers." In *SIGIR '05: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 168–175, 2000.

[KL05] Daniel Kelleher and Saturnino Luz. "Automatic hypertext keyphrase detection." In *IJCAI '05: Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 1608–1609, 2005.

[KZ04] Reiner Kraft and Jason Zien. "Mining anchor text for query refinement." In *WWW '04: Proceedings of the 13th International Conference on World Wide Web*, 2004.

[Lee04] Young-Suk Lee. "Morphological Analysis for Statistical Machine Translation." In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pp. 57–60, May 2 - May 7 2004.

[Lev66] Vladimir I. Levenshtein. "Binary Codes Capable of Correcting Deletions, Insertions and Reversals." In *Soviet Physics Doklady*, volume 10, pp. 707–710, 1966.

[LH99] Tessa Lau and Eric Horvitz. "Patterns of Search: Analyzing and Modeling Web Query Refinement." In *Proceedings of the Seventh International Conference on User Modeling*, 1999.

[LLC05] Uichin Lee, Zhenyu Liu, and Junghoo Cho. "Automatic identification of user goals in web search." In *WWW '05: Proceedings of the 14th International Conference on World Wide Web*, 2005.

[LYM02] Fang Liu, Clement Yu, and Weiyi Meng. "Personalized web search by mapping user queries to categories." In *CIKM '02: Proceedings of the 11th International Conference on Information and Knowledge Management*, pp. 558–565, 2002.

[LZ01] John Lafferty and Chengxiang Zhai. "Document language models, query models, and risk minimization for information retrieval." In *SIGIR '01: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 111–119, 2001.

[MMH08] Emily Moxley, Tao Mei, Xian sheng Hua, Wei ying Ma, and B. S. Manjunath. "Automatic video annotation through search and mining." In *ICME '08: Proceedings of the 2008 IEEE International Conference on Multimedia & Expo*, pp. 685–688, 2008.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[MWK06] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. "YALE: rapid prototyping for complex data mining tasks." In *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 935–940, 2006.

[MYK08] Hao Ma, Haixuan Yang, Irwin King, and Michael R. Lyu. "Learning latent semantic relations from clickthrough data for query suggestion." In *CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pp. 709–718, 2008.

[OKK02] B. Uygar Oztekin, George Karypis, and Vipin Kumar. "Expert agreement and content based reranking in a meta search environment using Mearf." In *WWW '02: Proceedings of the 11th International Conference on World Wide Web*, pp. 333–344, 2002.

[PBM98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The PageRank Citation Ranking: Bringing Order to the Web." Technical report, Stanford Digital Library Technologies Project, 1998.

[PCT06] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. "A Picture of Search." *The First International Conference on Scalable Information Systems*, June 2006.

[PG99] Alexander Pretschner and Susan Gauch. "Ontology Based Personalized Search." In *ICTAI '99: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, p. 391, 1999.

[PN] Xuan-Hieu Phan and Cam-Tu Nguyen. http://gibbslda.sourceforge.net/.

[Por97] Martin F. Porter. "An algorithm for suffix stripping." *Readings in information retrieval*, pp. 313–316, 1997.

[Pro] Open Directory Project. http://www.dmoz.org/.

[PV98] Massimiliano Pontil and Alessandro Verri. "Support Vector Machines for 3D Object Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**:637–646, 1998.

[QLC05] Feng Qiu, Zhenyu Liu, and Junghoo Cho. "Analysis of user web traffic with a focus on search activities." In *Proc. International Workshop on the Web and Databases (WebDB)*, pp. 103–108, 2005.

[RBG10] Sujith Ravi, Andrei Broder, Evgeniy Gabrilovich, Vanja Josifovski, Sandeep Pandey, and Bo Pang. "Automatic generation of bid phrases for online advertising." In *WSDM '10: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pp. 341–350, 2010.

[RCG05] Berthier Ribeiro-Neto, Marco Cristo, Paulo B. Golgher, and Edleno Silva de Moura. "Impedance coupling in content-targeted advertising." In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 496–503, 2005.

[RD06] Filip Radlinski and Susan Dumais. "Improving personalized web search using result diversification." In *SIGIR '06: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 691–692, 2006.

[Res] ABI Research. http://www.abiresearch.com/press/1138.

[Ris01] Irina Rish. "An empirical study of the naive Bayes classifier." In *IJCAI '01 workshop on "Empirical Methods in AI"*, 2001.

[RL04] Daniel E. Rose and Danny Levinson. "Understanding user goals in web search." In *WWW '04: Proceedings of the 13th International Conference on World Wide Web*, pp. 13–19, 2004.

[San08] Mark Sanderson. "Ambiguous queries: test collections need more sense." In *SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 499–506, 2008.

[SB88] Gerard Salton and Christopher Buckley. "Term-weighting approaches in automatic text retrieval." In *Information Processing and Management*, pp. 513–523, 1988.

[SG05] Micro Speretta and Susan Gauch. "Personalized Search Based on User Search Histories." In *WI '05: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 622–628, 2005.

[SG07] Mark Steyvers and Tom Griffiths. "Probabilistic Topic Models." In *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

[SH06] Mehran Sahami and Timothy D. Heilman. "A web-based kernel function for measuring the similarity of short text snippets." In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pp. 377–386, 2006.

[SLN09] Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. "Identification of ambiguous queries in web search." *Information Processing and Management*, **45**(2):216–229, 2009.

[SSS09] Stefan Siersdorfer, Jose San Pedro, and Mark Sanderson. "Automatic video tagging using content redundancy." In *SIGIR '09: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 395–402, 2009.

[SWG04] C. G. M. Snoek, M. Worring, J. M. Geusebroek, D. C. Koelma, and F. J. Seinstra. "The MediaMill TRECVID 2004 semantic video search engine." In *In TREC Video Retrieval Evaluation Online Proceedings*, 2004.

[Tur] Mechanical Turk. http://www.mturk.com/.

[Tur03] Peter D. Turney. "Coherent keyphrase extraction via web mining." In *IJCAI '03: Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 434–439, 2003.

[VH06] Atulya Velivelli and Thomas S. Huang. "Automatic Video Annotation by Mining Speech Transcripts." In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, p. 115, 2006.

[Voo94] Ellen M. Voorhees. "Query expansion using lexical-semantic relations." In *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 61–69, 1994.

[Voo04] Ellen M. Voorhees. "Overview of the TREC 2004 Robust Retrieval Track." In *Proceedings of the Thirteenth Text REtrieval Conference (TREC2004)*, p. 13, 2004.

[WC08] Michael J. Welch and Junghoo Cho. "Automatically identifying localizable queries." In *SIGIR '08: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 507–514, 2008.

[WCC10] Michael Welch, Junghoo Cho, and Walter Chang. "Generating Advertising Keywords from Video Content." In *CIKM '19: Proceeding of the 19th ACM Conference on Information and Knowledge Management*, 2010.

[Wik] Wikipedia. http://www.wikipedia.org/.

[WWX05] Lee Wang, Chuang Wang, Xing Xie, Josh Forman, Yansheng Lu, Wei-Ying Ma, and Ying Li. "Detecting dominant locations from search queries." In *SIGIR '05: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 424–431, 2005.

[WZ09] Jun Wang and Jianhan Zhu. "Portfolio theory of information retrieval." In *SIGIR '09: Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 115–122, 2009.

[WZC02] Chingning Wang, Ping Zhang, Risook Choi, and Michael D'Eredita. "Understanding consumers attitude toward advertising." In *Eighth Americas Conference on Information System*, pp. 1143–1148, 2002.

[XC96] Jinxi Xu and W. Bruce Croft. "Query expansion using local and global document analysis." In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 4–11, 1996.

[Xma] Xmarks. http://www.xmarks.com/.

[YGC06] Wen Tau Yih, Joshua Goodman, and Vitor R. Carvalho. "Finding advertising keywords on web pages." In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pp. 213–222, 2006.

[You] YouTube. http://www.youtube.com/.

[YRL09] Xing Yi, Hema Raghavan, and Chris Leggetter. "Discovering users' specific geo intention in web search." In *WWW '09: Proceedings of the 18th International Conference on World Wide Web*, pp. 481–490, 2009.

[ZCL03] Cheng Xiang Zhai, William W. Cohen, and John Lafferty. "Beyond independent relevance: methods and evaluation metrics for subtopic retrieval." In *SIGIR '03: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 10–17, 2003.

[ZWT09] Jianhan Zhu, Jun Wang, Michael Taylor, and Ingemar J. Cox. "Risk-Aware Information Retrieval." In *ECIR '09: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, pp. 17–28, 2009.

[ZZP08] S. Zanetti, L. Zelnik-Manor, and P. Perona. "A walk through the web's video clips." *CVPRW '08: Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 1–8, 2008.