

Foundations and Trends<sup>®</sup> in Information Retrieval  
Vol. XX, No. XX (2016) 1–139  
© 2016 J. Arguello  
DOI: 10.1561/XXXXXXXXXX



## Aggregated Search

Jaime Arguello  
School of Information and Library Science  
University of North Carolina at Chapel Hill  
jarguello@unc.edu

# Contents

---

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Aggregated Search Tasks . . . . .	4
1.2	Relation to Federated Search . . . . .	6
1.3	Differences between Aggregated and Federated Search . . . . .	8
1.4	Overview of Aggregated Search Algorithms . . . . .	10
1.5	Related Topics . . . . .	12
1.6	Related Surveys . . . . .	17
1.7	Outline . . . . .	18
<b>2</b>	<b>Sources of Evidence</b>	<b>20</b>
2.1	Typology of Features . . . . .	21
2.2	Notation . . . . .	23
2.3	Query Features . . . . .	23
2.4	Vertical Features . . . . .	27
2.5	Query-vertical Features . . . . .	27
2.6	Summary and Considerations . . . . .	37
<b>3</b>	<b>Approaches for Vertical Selection and Presentation</b>	<b>41</b>
3.1	Vertical Selection . . . . .	41
3.2	Vertical Presentation . . . . .	49
3.3	Summary . . . . .	56

<b>4</b>	<b>Evaluation</b>	<b>58</b>
4.1	Vertical Selection Evaluation . . . . .	59
4.2	End-to-end Evaluation . . . . .	68
4.3	Summary . . . . .	80
<b>5</b>	<b>Search Behavior with Aggregated Search</b>	<b>82</b>
5.1	Evaluation Metric Validation . . . . .	83
5.2	Studies Supporting Vertical Selection and Presentation . . . . .	84
5.3	Factors Affecting Vertical Results Use and Gain . . . . .	88
5.4	Spillover Effects in Aggregated Search . . . . .	91
5.5	Scanning Behavior in Aggregated Search . . . . .	95
<b>6</b>	<b>Special Topics in Aggregated Search</b>	<b>101</b>
6.1	Domain Adaptation for Vertical Selection . . . . .	101
6.2	Smoothing Vertical Click Data . . . . .	104
6.3	Composite Retrieval . . . . .	105
6.4	Query Disambiguation and Vertical Selection . . . . .	107
6.5	Aggregated Search for Children . . . . .	108
6.6	Aggregated Mobile Search . . . . .	110
<b>7</b>	<b>Conclusions</b>	<b>114</b>
7.1	Future Directions . . . . .	118
	<b>References</b>	<b>122</b>

## Abstract

The goal of aggregated search is to provide integrated search across multiple heterogeneous search services in a unified interface—a single query box and a common presentation of results. In the web search domain, aggregated search systems are responsible for integrating results from specialized search services, or verticals, alongside the core web results. For example, search portals such as Google, Bing, and Yahoo! provide access to vertical search engines that focus on different types of media (images and video), different types of search tasks (search for local businesses and online products), and even applications that can help users complete certain tasks (language translation and math calculations).

Aggregated search systems perform two main tasks. The first task (vertical selection) is to predict which verticals (if any) to present in response to a user's query. The second task (vertical presentation) is to predict where and how to present each selected vertical alongside the core web results.

The goal of this work is to provide a comprehensive summary of previous research in aggregated search. We first describe why aggregated search requires unique solutions. Then, we discuss different sources of evidence that are likely to be available to an aggregated search system, as well as different techniques for integrating evidence in order to make vertical selection and presentation decisions. Next, we survey different evaluation methodologies for aggregated search and discuss prior user studies that have aimed to better understand how users behave with aggregated search interfaces. Finally, we review different advanced topics in aggregated search.

# 1

---

## Introduction

---

In recent years, the field of information retrieval (IR) has broadened its scope to address a wide range of information-seeking tasks. Examples include search for images, video, news, digitized books, items for sale, local businesses, scholarly articles, and even social media updates such as tweets. A common finding in empirical IR research is that different information-seeking tasks require different solutions. Specifically, different tasks require different ways of representing items in the index, different retrieval algorithms for predicting relevance, and different ways of displaying search results to users.

Different types of media may require different representations. For example, images may need to be represented using text from the surrounding context in the originating page [Feng and Lapata, 2010], social media updates may need to be represented using text obtained from the link-to URL (if one is available) [McCreadie and Macdonald, 2013], and books may need to be represented using text from an external summary page [Koolen et al., 2009]. Different search tasks may also require customized retrieval algorithms. For example, news search may require favoring recently published articles [Diaz, 2009], local business search may require favoring businesses that are geographically close [Abou-

Assaleh and Gao, 2007], and scholarly article search may require favoring articles with many citations [Lawrence et al., 1999]. Finally, different search tasks may require different ways of presenting the search results to users, by highlighting the most important attributes of the underlying item. In current systems, for example, webpage results are typically displayed using the webpage title and a summary snippet showing the context where the query terms appear on the page; items for sale are typically displayed using a thumbnail image of the product, a description, and the price; and videos are typically displayed using a stillframe of the video, a description, and the duration.

Search systems today are more diverse and specialized than ever before. In fact, search portals that aim to support different information-seeking tasks typically develop and maintain specialized search systems for different task types. Rather than attempt to address all task types with a single monolithic system, the current trend is towards a “divide and conquer” approach. Naturally, this gives rise to a new challenge: How do we provide integrated search across these widely different systems? This is the goal of *aggregated search*. The aim of aggregated search technology is to provide integrated search across a wide range of highly specialized search systems in a unified interface—a single search query box and a common presentation of results.

To date, most research in aggregated search has focused on the web search domain. For this reason, most of the research reviewed in this article will also focus on the web search domain. Commercial web search portals such as Google, Bing, and Yahoo! provide access to a wide range of specialized search services besides web search. These specialized search services are referred to as *vertical search services* or simply *verticals*. Example verticals include search engines for different types of media (e.g., images, video, news) and search services for different types of search tasks (e.g., search for local business, products for sale, scientific articles). In some cases, search portals even provide access to verticals that help users accomplish specific tasks such as language translation, unit conversation, and math calculations.

There are currently two ways that users can access vertical content. If the user wants results from a specific vertical, and if the vertical has

direct search capabilities, then the user can issue the query directly to the vertical. In other cases, however, the user may not know that a vertical has relevant content, or may want results from multiple verticals at once. For this reason, an important task for commercial search providers has become the prediction and integration of relevant vertical content alongside the core web search results.

Figure 1.1 shows an example aggregated search results page (SERP) in the web domain. In response to the query “saturn”, an aggregated search system decided to display news, image, and video vertical results in addition to the core web results. The most confidently relevant verticals are displayed higher on the SERP. In this case, the system predicted that the most relevant verticals were the news, images, and video verticals, respectively.

## 1.1 Aggregated Search Tasks

Most aggregated search systems follow a pipeline architecture with three subsequent sub-tasks (Figure 1.2). The first sub-task (*vertical selection*) is to predict *which* verticals (if any) are relevant to the query. One can view the vertical selection task as that of deciding which verticals should be displayed on the SERP regardless of their position. It is impractical, if not impossible, to issue the query to every available vertical. For this reason, most approaches for vertical selection base their predictions using *pre-retrieval* evidence (e.g., the query contains the term “news”, the query is related to the health domain, or the query contains the name of a location).

The second sub-task (*vertical results selection*) is to predict which results from a particular vertical to present on the aggregated SERP. This sub-task has received the least attention in the research community. The vertical results selection task has a dual objective. The primary objective is to satisfy the user directly with the vertical results that are aggregated on the SERP. The secondary objective is more nuanced. Some verticals have direct search capabilities. If the user realizes that the vertical may have relevant information, he or she can navigate to the vertical, examine more vertical results, and even issue

query saturn

**news**

**News for saturn**

**PHOTO: Saturn's Holiday Closeup**  
 NPR (blog) - by Mark Memmott - 17 hours ago  
 NASA's Cassini spacecraft focused on one of the planet's poles, and produced an image that resembles a hand-painted Christmas ornament.

**Best New Space Pictures: Saturn's Crown and Astronauts' Renown**  
 National Geographic - 17 hours ago

**Saturn dazzles in new NASA images**  
 San Jose Mercury News - 3 days ago

**Saturn: Cars, SUVs & Crossover Vehicles**  
 www.saturn.com/ -  
 The official site for Saturn cars, suvs & crossovers. Find warranty information, locate a service dealer, or view Saturn models including the VUE, Outlook, Aura, ...  
 Saturn Cars, SUVs ... - Locate Saturn Service Dealers - Owner Resources

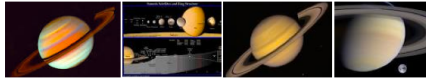
**web**

**Saturn - Wikipedia, the free encyclopedia**  
 en.wikipedia.org/wiki/Saturn -  
 Saturn is the sixth planet from the Sun and the second largest planet in the Solar System, after Jupiter. Named after the Roman god of agriculture, Saturn, its ...  
 Saturn (mythology) - Saturn (disambiguation) - Rings of Saturn - Moons of Saturn

**Saturn (mythology) - Wikipedia, the free encyclopedia**  
 en.wikipedia.org/wiki/Saturn\_(mythology) -  
 Saturn (Latin: Saturnus) was a god in ancient Roman religion, and a character ...

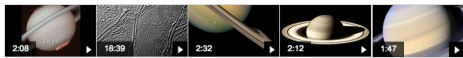
**images**

**Images for saturn** - Report images



**videos**

**Videos of saturn**



Saturn's Beautiful Aurora YouTube  
 Saturn's Mysterious Mo... YouTube  
 5.6k Saturn Cassini Photog... vimeo  
 The Planet Saturn Discovery  
 The Mystery Hexagon on SA... YouTube

...

**Figure 1.1:** Aggregated SERP in the web domain (truncated). In response to the query “saturn”, the aggregated search system decides to display news, image, and video vertical results in addition to the core web results. The most confidently relevant verticals are displayed higher on the SERP.

new queries to the vertical search engine. In this respect, the secondary objective of vertical results selection is to convey how the underlying vertical may have relevant content. Most aggregated search systems described in the published literature do not perform vertical results selection and simply display the top few results returned by the vertical in response to the query.

The third and final sub-task (*vertical presentation*) is to decide *where* to present each selected vertical. Different verticals are typically associated with different surrogate representations. For example, image results are displayed using thumbnails, while news results are displayed using the article title, source, publication date, and may include an op-



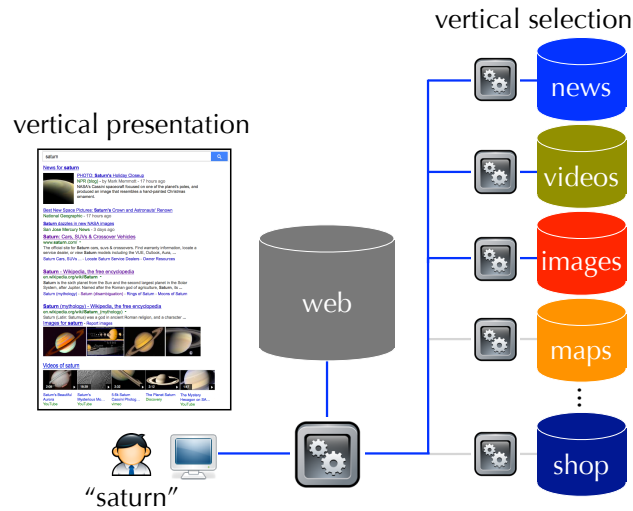


Figure 1.2: Aggregated search pipeline.

tional image from the underlying article. For aesthetic reasons and to better convey how the vertical may have relevant content for the current user, vertical results are typically grouped together (either stacked horizontally or vertically) on the aggregated SERP.

The goal of vertical presentation is to display the most relevant verticals in a more salient way. One common approach is to display them higher on the SERP (e.g., above the first web result). Vertical presentation happens after the query has been issued to the vertical. Thus, approaches for vertical presentation can base their predictions using pre-retrieval as well as *post-retrieval* evidence (e.g., the number of results returned by the vertical, the top retrieval scores, or the number of query-terms appearing in the top results).

## 1.2 Relation to Federated Search

While aggregated search may seem like a new technology, it is rooted in a fairly mature subfield in information retrieval known as *federated search* or *distributed information retrieval*. The goal of *federated search* is to provide integrated search across multiple collections of *textual*

documents, also referred to as *resources*. Similar to aggregated search, federated search is typically decomposed into three sub-tasks.

The first sub-task (*resource representation*) is to construct a description of each distributed resource that can be used to predict which ones to search in response to a query. Approaches for resource representation differ greatly depending on whether they assume a *cooperative* or *uncooperative* environment. In a cooperative environment, resources are assumed to readily publish term statistics that can be used to model the contents of each collection [Gravano et al., 1997]. On the other hand, in an uncooperative environment, resources are assumed to only provide a search interface. In this case, resource descriptions must be constructed from sampled documents obtained via query-based sampling. In general, query-based sampling involves issuing queries to each resource and downloading results [Callan and Connell, 2001b; Caverlee et al., 2006; Shokouhi et al., 2006a].

The second sub-task (*resource selection*) is to predict which resources to search in response to a query. Typically, the relevant documents are concentrated in only a few of the available resources. Resource selection approaches tend to cast the task as *resource ranking*—ranking resources based on the likelihood that they will return relevant results for the query. Existing approaches can be categorized into two types: *large document* and *small document* models. Large document models select resources based on the similarity between the query and a virtual concatenation of all the documents in the resource (or its samples). These methods treat each collection as a large document and adapt document-ranking algorithms for the purpose of ranking collections. In contrast, small document models typically proceed in two steps. First, they combine documents (or samples) from the different resources in a *centralized sample index* (CSI). Then, at query-time, they rank resources based on the top-ranked CSI results [Si and Callan, 2003a; Shokouhi, 2007; Thomas and Shokouhi, 2009].

The third sub-task (*results merging*) is to interleave the results from the different selected resources into a single ranking. Typically, this is cast as a score normalization problem [Si and Callan, 2003b]. Because different resources have different collection statistics and per-

haps use different ranking algorithms, their retrieval scores may not be directly comparable. Thus, results merging requires transforming resource-*specific* scores into resource-*agnostic* scores that can be used to produce a single merged ranking. Results merging approaches typically assume that documents can be interleaved in an unconstrained fashion. The only goal is to rank the relevant documents higher on the list, irrespective of the originating resource(s).

Most federated search approaches make assumptions that do not hold true in an aggregated search environment. Thus, while there are similarities between aggregated and federated search, aggregated search requires unique solutions. Next, we discuss some of the main difference between the aggregated and federated search.

### 1.3 Differences between Aggregated and Federated Search

**Cooperative vs. uncooperative environment.** Most federated search approaches assume an uncooperative environment in which the different resources provide the system no more than the same functionality they provide their human users—a search interface. For this reason, most resource selection approaches base their predictions solely on the similarity between the input query and the documents sampled from each resource. In contrast, most aggregated search approaches assume a cooperative environment in which the different verticals are developed and maintained by the same organization. In a cooperative environment, the aggregated search system may have access to sources of evidence beyond sampled documents. For example, for verticals with direct search capabilities, alternative sources of evidence may include vertical-specific query-traffic data, click-through data, and query-reformulation data. This type of evidence conveys how users interact directly with the vertical search engine and may be helpful in predicting vertical relevance. A vertical selection system should be capable of incorporating these various sources of evidence into selection decisions.

**Heterogeneous vs. homogeneous content.** Most federated search approaches assume that all the distributed resources contain

*textual* documents. For example, small document approaches for resource selection assume that samples from different resources can be combined in a centralized sample index (CSI), and that resources can be selected based on the top-ranked CSI results. In contrast, approaches for vertical selection need to accommodate the fact that different verticals may contain very different types of items that can not be centrally indexed and searched (e.g., news articles, images, videos, items for sale, digitized books, social media updates, etc.).

**Heterogeneous vs. homogeneous relevance prediction.** Most federated search approaches apply the *same* scoring function to *every* available resource in order to predict their relevance to a query. For example, small document approaches score every resource based on the top CSI results. Similarly, large document models score every resource based on the similarity between the query and a virtual concatenation of those documents sampled from the resource. In contrast, approaches for vertical selection and presentation must be able to learn a *vertical-specific* relationship between different types of evidence and a particular vertical’s relevance to a query.

To illustrate, let us consider two examples. First, certain key words are likely to predict that a particular vertical is relevant to the query. For example, the query term “news” suggests that the news vertical is relevant, while the query term “pics” suggests that the images vertical is relevant. Second, some verticals tend to be topically focused (e.g., health, auto, travel, movies). Thus, in some cases, it may be possible to predict that a particular vertical is relevant based on the general topic of the query. For example, we can predict that the health vertical is relevant to the query “swine flu” because the query is related to the health domain. Both of these examples suggest that aggregated search approaches must be able to learn a *vertical-specific* relation between certain types of evidence and the relevance of a particular vertical.

**Selection vs. ranking** Most federated search approaches treat resource *selection* as resource *ranking*. The goal for the system is to prioritize resources in response to a query, and to select as many or as few resource as possible given the current computational resources available. Implicit in this formulation of the resource selection task is

the assumption that *exhaustive* search produces a good retrieval and that the goal for the system is to approximate this retrieval by selecting only a few resources. In contrast, vertical selection requires predicting which verticals are relevant to the query and which verticals are not. In some cases, the system may decide that none of the available verticals are relevant. Thus, vertical selection requires approaches that can make binary predictions for each candidate resource.

**Constrained vs. Unconstrained Results Presentation.** Finally, most federated search approaches assume that the results from the different selected resources can be interleaved in an unconstrained fashion. In contrast, most aggregated search approaches assume that the results from the same vertical must be presented together on the SERP in the form of a vertical block. This is mostly done for aesthetic reasons and to provide an easy-to-parse overview of how the vertical may have relevant content for the query. Vertical presentation approaches must address the unique challenge of deciding where to present each selected vertical on the SERP.

## 1.4 Overview of Aggregated Search Algorithms

Most successful approaches for vertical selection and presentation use machine learning to combine a wide range of evidence as input features to the model. Features can be generated from the query, from the vertical, or from the query-vertical pair. For example, a type of query feature might consider whether the query contains the keyword “news”, a type of vertical feature might consider the number of recent clicks on the vertical results, and a type of query-vertical might estimate the number of query-related documents in the underlying vertical collection. The most effective approaches for vertical selection and presentation make creative use of the different sources of evidence available to the system, including vertical-specific query-log data, sampled vertical documents, and previous user interactions with vertical content.

While evidence integration is key to aggregated search, it also poses two main challenges. The first challenge is that not all features may be available for all verticals. For example, some verticals cannot be directly

searched by users. Consider the weather vertical in most commercial search portals. Users cannot typically go directly to the weather vertical and issue a query. Thus, features generated from the vertical query-log will not be available for verticals that are not directly searchable. Similarly, some verticals are not associated with an underlying collection of documents. Consider the calculator, language translation, and finance verticals in most commercial search portals. Features that consider the similarity between the query and the documents in the underlying vertical will not be available for such verticals. In this respect, approaches for vertical selection and presentation must deal with the fact that different verticals may require different feature representations.

The second challenge is that, even if a feature is available for all verticals, it may not be *equally* predictive across verticals. For example, certain verticals are clicked more than others. For example, a news vertical is likely to have more clicks than a weather vertical, which is designed to display the necessary information directly on the SERP. Features derived from click data (e.g., the number of recent clicks on the vertical results) may be more predictive for verticals that have more clicks. Alternatively, a feature may be *positively* predictive for one vertical and *negatively* predictive for another. Consider, for example, a feature that measures whether the query is related to the travel domain. This feature is likely to be positively predictive for a travel-related vertical, but negatively predictive for a vertical that focuses on a different domain. In this respect, approaches for vertical selection and presentation must deal with the fact that different verticals may require learning a *vertical-specific* relationship between certain features and a vertical's relevance.

Given the two challenges outlined above, approaches for vertical selection typically learn a different model for each candidate vertical. In this way, each model can adopt a different feature representation and can learn a vertical-specific relationship between feature values and the relevance of the particular vertical. Vertical presentation requires resolving contention between different verticals to be displayed on the SERP. Put differently, vertical presentation requires predicting the degree of relevance of a vertical relative to the web results and relative

to other verticals to be displayed. Approaches for vertical presentation can be categorized into two types: pointwise and pairwise interleaving methods. Pointwise methods learn to predict the *degree* of relevance of each vertical block or module in response to a query. Vertical blocks are positioned according to their predicted relevance to the query. Pairwise methods learn to predict the relative relevance between *pairs* of vertical and/or web blocks or modules. Vertical blocks are positioned such that they are maximally consistent with the pairwise preferences predicted by the system.

## 1.5 Related Topics

In this review, we focus on aggregated search in the web domain, where systems combine results from heterogeneous sources (or verticals) into a single presentation. We cover a wide range of topics, including prediction, evaluation, and studies of user behavior.

We focus on the web domain because of most of the published research has been done in this domain. However, the task of searching and integrating information from heterogeneous sources happens in other domains within the broad field of information retrieval. For example, in desktop search, the system needs to search across different types of files, which may require different indexing structures, ranking algorithms, and ways of presenting the search results. Similarly, news aggregators are responsible for combining content from different input streams, such as news articles, images, videos, and social media updates.

In this section, we briefly describe related areas of IR research that may benefit from the algorithms, evaluation methods, and studies described in this review. These areas are described upfront in the interest of readers who may not have a primary interest in aggregated search in the web domain.

### 1.5.1 Full-text Search in Peer-to-Peer Networks

A peer-to-peer (P2P) network is defined as a network of independent computing resources that do not require a centralized authority to co-

ordinate and perform tasks. A *hierarchical* (P2P) network is one with three types of peers: (1) peers that provide search for a particular collection, such as a digital library (*providers*), (2) peers that originate information requests for the network (*consumers*), and (3) peers that propagate information requests to neighboring peers and send results back the corresponding consumer (*hubs*). Hubs perform the three main tasks associated with aggregated search: (1) representing the contents of neighboring peers (i.e., direct providers and other hubs), (2) sending information requests to the neighboring peers most likely to deliver relevant content, and (3) merging the results returned by the selected peers and sending these back to the appropriate consumer. Lu [2007] proposed several approaches for these three different tasks that build upon traditional federated search techniques (where there is a centralized federated search system that has direct access to all available resources).

The techniques discussed in this review might be useful for the tasks of query routing and results merging in P2P networks that provide distributed search capabilities. Beverly and Afegan [2007], for example, proposed a machine learning, evidence integration approach for neighbor selection in P2P networks.

### 1.5.2 Desktop Search

The goal of *desktop search* is to facilitate search over files stored in a user's desktop computer. One of the main challenges in desktop search is that different file types are associated very different field structures and meta-data. Kim and Croft [2010] developed and evaluated a desktop search system that maintains different indexes for different file types. Given a query, the proposed system performs the three basic steps associated with aggregated search: file-type prediction, file-type-specific ranking, and results merging. Much like the vertical selection methods covered in this review, the proposed file-type prediction approach combined multiple types of evidence as features for a machine learned model, for example, the similarity between the query and document meta-data, the similarity between the query and previously run queries with clicks on a particular file-type, and the presence of certain



query keywords such as “email” or “pdf”. As one might expect, the evidence integration approach to file-type prediction outperformed the best approach using a single source of evidence.

### 1.5.3 Selective Search

The aim of *selective search* is to enable efficient and effective search from large text collections in environments with modest computational resources [Kulkarni and Callan, 2015]. First, the system partitions the large text collection into smaller *topical* sub-collections or *shards*. Then, in response to a query, the system predicts which few shards are most likely to have relevant documents and merges their results. Selective search is highly motivated by the *cluster hypothesis*, which states that similar documents (ideally assigned to the same shard) tend to be relevant to same information needs [Rijsbergen, 1979]. Shard representation and selection can be performed using existing federated search techniques, and results merging is relatively straightforward because the system has access to global term statistics can be used to compute comparable retrieval scores. The critical step in selective search is partitioning the collection into topical shards. Kulkarni and Callan [Kulkarni and Callan, 2015] proposed a variant of the well-known K-means clustering algorithm that operates on a sample of documents from the collection. Experimental results show that selective search can greatly reduce computational costs and latency, and can yield retrieval performance comparable to exhaustive search, particularly for precision-oriented tasks.

While current shard-selection techniques do not combine multiple types of evidence to make predictions, prior work on text-based federated search used machine learning to combine a wide range of features for the task of resource selection [Arguello et al., 2009a; Hong et al., 2010]. In particular, because shards are topically focused, the query category features discussed later in Section 2.3 might contribute valuable evidence for shard selection.

#### 1.5.4 Contextual Suggestion

The goal of *contextual suggestion* is to recommend points-of-interest (POIs) to a user in a particular context (i.e., in a particular location, at a particular time) [Dean-Hall et al., 2012, 2013, 2014, 2015]. The system is assumed to have access to ratings on previously recommended POIs for the same user (or to other users) in different contexts.

Zhuang et al. [2011] describe a mobile contextual suggestion system with an aggregated search architecture. Rather than index and retrieve all POIs using a single system, the proposed approach is to build different indexes and rankers for different POI-types (e.g., restaurants, coffee shops, bars, tourist attractions, etc.) The system recommends POIs to a user in a particular context in two steps. First, the system predicts the appropriateness of a particular POI-type for the given context, and then it ranks POIs of a particular type if the user requests to see those results. Similar to aggregated search, the proposed architecture has two main benefits. First, the system can use different models for predicting relevance for each POI-type. For example, the system can learn that restaurants are more relevant during meal times and that bars are more relevant in the evening. Second, the system can learn different rankers for different POI-types. For example, the system can determine that close proximity to the user is more important for coffee shops than for tourist attractions (assuming users are more willing to travel longer distances for the latter).

#### 1.5.5 Search Across Heterogeneous Social Networks

In certain cases, a user may engage in multiple social networks and may want to receive updates from different networks in a unified interface. Bian et al. [2012] proposed an algorithm for ranking social network updates originating from different networks. The main challenge is that different networks may be associated with different sources of evidence that can be used to predict the relevance of an update for a particular user. Consider a user who wants to receive aggregated updates from both Facebook and Twitter. Some sources of evidence are common to both networks (e.g., Does the update contain a URL?). However, other

features may aim to exploit the same type of evidence, but be associated with very different numerical ranges across networks (e.g., number of comments on Facebook and number of retweets on Twitter). Moreover, some features may only exist in one network and not the other (e.g., the number of Facebook chat messages between the user and the author of the update). Rather than rank candidate updates from different networks using a single model (perhaps using only those features common to all networks), Bian et al. [2012] describe a “divide and conquer” approach that learns network-specific rankers and combines their output rankings into a single ranked list.

Lee et al. [2012] focused on the task of ranking social media updates and used two test collections: one generated from Facebook updates and another generated from Twitter updates. The authors did not attempt the task of constructing a single, merged ranking. However, the authors concluded that combining updates from different heterogeneous social networks into a single ranked list is an interesting research direction for future work.

### 1.5.6 News Aggregators

News content aggregators such as the Yahoo! homepage or the New York Times homepage combine results from different heterogeneous data streams into a single presentation. Data streams may include news articles from different sources, images, videos, audio interviews, blog posts, and social media updates such as tweets. The system is responsible for predicting which items to display from each data stream and where [Bharat et al., 1998; Krakovsky, 2011]. Different data streams are likely to be associated with very different types of evidence that can be used to predict relevance. Thus, news aggregators are likely to benefit from a “divide and conquer” approach—building customized rankers for different data streams and a system that predicts which content to display and where.

One interesting aspect of news aggregation is that in some cases, the system may want to show results from different data streams that are related to the same topic. For example, the system may want to display news, images, videos, and opinionated tweets about the same trending

news story. Hong et al. [2011] proposed an approach for finding related content in different data streams. In the context of aggregated search, the results from different sources aggregated on the search results page are typically independent of each other. However, identifying related results may be an interesting direction for future work.

## 1.6 Related Surveys

As mentioned above, aggregated search is related to the subfield of federated search or distributed information retrieval, where the goal is to provide integrated search across multiple *textual* collections. Shokouhi and Si [2011] provide an extensive review of the state of the art in federated search, and review methods for all three federated search sub-tasks: resource representation, selection, and results merging.

Chapter 4 in this review focuses on methods of aggregated search evaluation. Online evaluation approaches learn about a system's performance from user interactions in a live environment. In the context of aggregated search, vertical selection approaches can be evaluated by considering user's clicks on the vertical results. Interpreting user interactions with a SERP is complicated by the fact that users are biased by factors that are independence of relevance, such as position and visual salience. Katja Hofmann [2016] provide an extensive survey of approaches for online evaluation using real users.

The current survey is most closely related to the book chapter titled "Aggregated Vertical Search" appearing in Long and Chang [2014]. However, the current survey is different in several respects. First, it includes new solutions, evaluation methods, and user studies published since 2014. In recent years, studies have proposed and tested new evaluation metrics for aggregated search [Zhou et al., 2013b]. Furthermore, recent studies have investigated different factors that may affect search behavior and performance with aggregated search interfaces. For example, recent work investigates how users visually scan an aggregated SERP [Liu et al., 2015], how the results from one source on the SERP can affect user interaction with the results from other sources [Arguello and Capra, 2016; Bota et al., 2016], and how users' cognitive abilities

can affect different search behaviors and outcomes [Turpin et al., 2016].

Furthermore, this review covers more special topics in aggregated search. For example, it surveys recent work on *composite retrieval*, where the goal for the system is to combine results from different sources, but to organize them by how they satisfy different *aspects* of the user’s task. Also, it covers recent work on aggregated search for children, who tend to exhibit different search behaviors than adults and require unique aggregated search solutions [Duarte Torres and Weber, 2011a].

## 1.7 Outline

As previously mentioned, the most effective approaches for vertical selection and presentation use machine learning to combine different types of evidence as features. Chapter 2 reviews different features used in prior work. These include features that derive evidence from vertical content, from queries issued directly to the vertical by users, and from previous users’ interactions with the results from a particular vertical.

In a sense, vertical selection and presentation have a common goal—to predict the degree of relevance of a vertical to a user’s query. In Chapter 2, we remain somewhat agnostic as to whether a particular feature is more appropriate for one task versus the other. That said, certain features (referred to as *post-retrieval* features) require issuing the query to the candidate vertical. Thus, in some places, we emphasize that post-retrieval features may be more appropriate for vertical presentation.

Chapter 3 focuses on evidence combination approaches for vertical selection and presentation. The main challenge in vertical selection and presentation is that certain features may be predictive for one vertical, but not another. For example, the publication age of the top vertical results may be predictive for the news vertical, but not the image vertical. Moreover, certain features may be *positively* predictive for one vertical, but *negatively* predictive for another. For example, the query term “news” is positively predictive for the news vertical, but negatively predictive for the image vertical. For this reason, in Chapter 3

we focus on approaches that can learn and exploit a *vertical-specific* relationship between different features and the relevance of a particular vertical.

Chapter 4 focuses on evaluation methodologies and metrics for aggregated search. Evaluation is a critical component of all information retrieval techniques and a research area in its own right. We start with vertical selection and then cover end-to-end evaluation, which includes selection and presentation. We cover evaluation methodologies based on re-usable test collections, which typically include a set of evaluation queries, cached results from the different sources, and human-produced relevance judgements. We also discuss on-line evaluation methodologies based on implicit feedback from real users in an operational setting.

Chapter 5 reviews user studies aimed at further understanding what users want from an aggregated search system and how they behave. We cover studies where the goal is to determine the extent to which a particular evaluation metric correlates with user satisfaction, and studies where the goal is to understand how different characteristics of the interface, the search task, and the user can affect outcome measures associated with the user's perceptions about the system and their performance.

Chapter 6 reviews special topics in aggregated search. Here, we touch upon algorithms for predicting how a user will visually scan a particular aggregated SERP, methods for obtaining implicit feedback that can improve prediction performance, and approaches for learning a model for a new vertical with little human-produced training data. Furthermore, we review the new task of *composite retrieval*, where the goal is to organize results from different sources based on different *aspects* associated with the task. Finally, we discuss aggregated search for children, who exhibit different behavior than adults and require unique solutions.

Finally, in Chapter 7, we conclude by highlighting the main trends in aggregated search and discussing short-term and long-term areas for future work.

# 2

---

## Sources of Evidence

---

State-of-the-art methods for vertical selection and presentation combine a wide range of evidence to make predictions. A convenient way of combining evidence is to train a model using machine learning. Machine learning algorithms learn to make predictions using a set of positive and negative examples. For instance, we can imagine learning a vertical selection model for a news vertical using a set of example queries for which the system should and should not select the news vertical. The system designer, however, is responsible for deciding how to represent query-vertical pairs using a set of measures or *features*. Good features are those that are highly correlated with the vertical's relevance to a query and bad features are those that are uncorrelated. A lot of creativity goes into designing effective features.

There are many ways in which an aggregated search system might predict that a particular vertical is relevant to a query. Consider the task of predicting whether a news vertical is relevant. If the query contains the term “news”, it is almost certain the news vertical is relevant. Similarly, a system might determine that the news vertical is relevant to the query “presidential election” because many of the documents in the news collection contain these query terms. Finally, a system might

predict that the news vertical is relevant if the query is similar to a recent burst of queries issued directly to the news vertical by users. The most successful approaches for vertical selection and presentation use machine learning to combine a wide-range of evidence as input features to a model.

In this chapter, we describe the most effective features used in prior research. In learning about different types of features, it is helpful to be aware of their similarities and differences. For example, it is helpful to understand the resources required for generating each feature, and whether generating a feature requires issuing the full query to a particular vertical. We begin the chapter with a description of two dimensions along which predictive features can be categorized. Then, we describe different types of features used in prior work and their implementation details.

## 2.1 Typology of Features

Features can be characterized along two dimensions. The first dimension relates to whether the value of the feature depends on the input query, the vertical under consideration, or the query-vertical pair. *Query features* describe properties of the input query and their values are independent of the vertical under consideration. A type of query feature might consider whether the query contains the keyword “news”. *Vertical features* describe properties of a particular vertical and their values are independent of the query. A type of vertical feature might consider the number of queries issued to the vertical directly by users in the recent past, which is a measure of the vertical’s current popularity. Finally, *query-vertical features* describe properties of the specific query *and* the vertical under consideration. A type of query-vertical feature might consider the number of results returned by the vertical in response to the query, which suggests that the vertical has a large amount of content related to the query topic.

Characterizing features along this first dimension is helpful in understanding the role of machine learning for vertical selection and presentation. Vertical and query-vertical features tend to have a *consistent*



relationship between the feature value and the relevance of the vertical in question (i.e., the vertical from which the feature value was derived). For example, a sudden burst in query traffic (a type of vertical feature) is likely to contribute positive evidence for any vertical under consideration. Similarly, the number of documents in the vertical containing all the query terms (a type of query-vertical feature) is likely to contribute positive evidence for any vertical.

In contrast to vertical and query-vertical features, query features tend to have an *inconsistent* relationship with vertical relevance across different verticals. In this respect, query features pose a unique challenge for vertical selection and presentation. For example, the presence of the query term “news” is positive evidence for the news vertical, but *negative* evidence for the images vertical. Similarly, the presence of the query term “pics” is positive evidence for the images vertical, but *negative* evidence for the news vertical. Machine learning provides a convenient way of learning a *vertical-specific* relationship between query features and the relevance of a particular vertical. For example, a common approach is to learn different models for different verticals. In doing so, each vertical-specific model can focus on the features that are uniquely predictive for the vertical in question.

The second dimension along which to characterize features relates to whether generating the feature value requires issuing the query to the vertical under consideration. *Pre-retrieval* features can be generated without issuing the query to the vertical. Query features and vertical features tend to be pre-retrieval features. For example, a type of pre-retrieval feature might describe the topical category of the query (a type of query feature) or the number of queries recently issued directly to the vertical by users (a type of vertical feature). In contrast, *post-retrieval* features must be generated by issuing the query to the vertical. For example, a type of post-retrieval feature might consider the average publication age of the top vertical results.

Characterizing features along this second dimension is helpful in understanding their usefulness for the tasks of vertical selection and/or vertical presentation. As previously mentioned, it is oftentimes impractical, if not impossible, to issue the query to every vertical in order to

decide which verticals to display. Thus, vertical selection approaches typically use only pre-retrieval features to make predictions. On the other hand, vertical presentation approaches typically assume access to post-retrieval evidence that can be used to decide the *degree* of relevance of a vertical to a query.

## 2.2 Notation

We will describe features using the following notation. Vertical selection and presentation requires predicting the relevance of a vertical to a query. We use  $q$  and  $v$  to denote the query and candidate vertical in question. In some cases, it is important to normalize feature values by comparing across *all* candidate verticals. For example, consider a feature that measures the probability of query  $q$  from a language model derived from vertical  $v$ 's documents. Such a feature might be more effective if we consider its value *relative* to the other candidate verticals. In such cases, we will  $\mathcal{V}$  to denote the set of all candidate verticals.

Finally, we use  $\phi^\star$  to denote a vector of features of type  $\star$ . Query features are denoted by  $\phi_q^\star$ , vertical features are denoted by  $\phi_v^\star$ , and query-vertical features are denoted by  $\phi_{q,v}^\star$ .

Using this notation, we can think of a query-vertical pair as a vector of features that describe different attributes of the query, the candidate vertical, and the query-vertical pair:

$$\left[ \phi_q^\star \dots \phi_v^\star \dots \phi_{q,v}^\star \right]$$

## 2.3 Query Features

Query features are generated from the query and are independent of the vertical being considered.

**Query string features.** Query string features consider the presence or absence of certain keywords appearing in the query [Arguello et al., 2009b, 2010; Diaz and Arguello, 2009; Jie et al., 2013; Li et al., 2008; Tsur et al., 2016; Wang et al., 2016]. For example, query terms such as “images”, “pictures”, and “pics” suggest the image vertical is

relevant, while query terms such as “buy”, “price”, and “shop” suggest the shopping vertical is relevant.

Tsur et al. [2016] also considered certain parts-of-speech appearing in the query for the task of predicting relevance for a community question-answering (CQA) vertical.

Query string features are typically binary-valued.

**Query characteristic features.** Query characteristic features describe attributes of the query, such as the query length, capitalization, or the presence of certain characters [König et al., 2009; Ponnuswami et al., 2011b,a; Tsur et al., 2016]. For example, a long query ending in a question mark may suggest that the CQA vertical is relevant, while capitalization may indicate the presence of a named entity and may suggest that the news vertical is relevant.

Prior work has also considered the presence of a particular named-entity type (e.g., person, location, organization) [Arguello et al., 2009b; Diaz and Arguello, 2009; Arguello et al., 2010, 2011a]. For example, the presence of a city name may suggest that the maps vertical is relevant, while the presence of a company name may suggest that the finance vertical is relevant.

**Query class features.** Queries can be categorized into different classes associated with the user’s intent. For example, Broder [2002] characterized web queries into three classes: navigational (the intent is to reach a specific page), informational (the intent is to find information present in one or more pages), and transactional (the intent is to perform a web-mediated transaction). Certain query classes are more likely to benefit from vertical results than others. For examples, information queries are more likely to benefit from vertical results than navigational queries.

In the context of aggregated search, prior work considered features associated with the likelihood of the query being a navigational query (presumably one for which the system should not display vertical results). König et al. [2009] used simple regular expressions to determine whether the query contains a URL. Ponnuswami et al. [2011b] used a proprietary query classifier. While details of the query classifier are not described in the paper, one could imagine that navigational queries are

frequent, have a low *click entropy* (i.e., are associated with clicks on the same result(s)), and have a high term-overlap with the title of the most clicked result(s). Jansen et al. [2008] discusses several simple heuristics for identifying navigational, informational, and transactional queries.

**Query category features.** One of the most effective query features used in prior work are query category features, which measure the the query’s affinity to a pre-defined set of topical categories [Arguello et al., 2009b,a, 2010, 2011a; Diaz and Arguello, 2009; Ponnuswami et al., 2011b,a; Wang et al., 2016]. Query category features have been successful for two reasons. First, many of the verticals investigated in prior vertical selection and presentation research have been topically focused (e.g., finance, health, movies, sports, travel). Second, query categorization has been studied widely in the context of other information retrieval tasks such as document ranking [Bennett et al., 2010] and ad matching [Broder et al., 2007]. Thus, aggregated search systems can make use of well-tested query categorization approaches for the purpose of feature generation.

Query categorization is challenging because state-of-the-art classifiers tend to use a bag-of-words representation and queries are usually very terse. A simple and effective solution is to categorize the query *indirectly* by issuing the query to a collection of pre-categorized documents and classifying the query based on the categories associated with the top search results [Shen et al., 2006].

Let  $\mathcal{C}$  denote the set of topical categories under consideration (e.g., finance, health, movies, sports, travel, etc.) and let  $\mathcal{R}_q^n$  denote the top- $n$  search results returned in response to query  $q$  from the collation of pre-categorized documents. The affinity of query  $q$  to category  $c \in \mathcal{C}$  can be computed as:

$$P(c|q) = \frac{1}{\mathcal{Z}} \sum_{d \in \mathcal{R}_q^n} P(c|d) \times score(d, q),$$

where  $P(c|d)$  denotes the prediction confidence value that document  $d$  belongs to category  $c$ ,  $score(d, q)$  denotes the retrieval score of document  $d$  in response to query  $q$ , and the normalizing factor  $\mathcal{Z} = \sum_{d \in \mathcal{R}_q^n} score(d, q)$ .

In the above formula,  $P(c|q)$  is proportional to the average affinity of

documents in  $\mathcal{R}_q^n$  with respect to category  $c$ . However, the average is a *weighted* average, where the weights are associated with each top document’s mass-normalized retrieval score. This is analogous to how we estimate the probability of terms in a *relevance language model* [Lavrenko and Croft, 2001]

Arguello et al. [2009b] conducted a feature ablation study and found that removing query category features caused the largest drop in vertical selection performance. Naturally, many of the verticals considered in this study were topically focused (e.g., travel, health, games, music, autos, sports, movies, finance, etc.).

König et al. [2009] focused on the task of vertical selection for the news vertical and computed query category features using a different approach. In this case, the authors computed different query-term statistics (e.g., average collection term frequency) from three different collections. Two collections were intended to represent newsworthy topics: a collection of recently published blogs and a collection of recently published news articles. The third collection (i.e. Wikipedia) was intended to represent non-newsworthy topics. The query was classified as belonging to the “news” category by comparing query-term statistics such as the average collection term frequency (averaged across query terms) between the different collections.

**Query ambiguity features.** Given an ambiguous or underspecified query, a common strategy for a search system is to diversify its results. Prior research in aggregated search has not considered query ambiguity as a source of evidence. However, it seems plausible that presenting results from different verticals might be one way to address the different possible intents of the user. Luo et al. [2014] describe a query ambiguity classifier and mention vertical selection as one possible down-stream application that may benefit from knowing about query ambiguity. The authors combine different features derived from the query string (e.g., the query length), click information from previous impressions of the same query (e.g., the click entropy), and the topical diversity associated with same-session queries in a query-log.

## 2.4 Vertical Features

Vertical features are derived from the vertical and are independent of the input query. In document retrieval, *document priors* are typically used to favor certain documents over others *irrespective* of the query. In the context of aggregated search, a vertical feature can be viewed as a type of prior probability that the vertical is relevant to a query. Broadly speaking, vertical features measure the current demand for a particular vertical. A greater demand suggests that the vertical is relevant.

Vertical features have not been frequently used in prior aggregated search research. Two studies used the click-through rate associated with previous presentations of the vertical for *any* query [Jie et al., 2013; Wang et al., 2016]. We could imagine using other measures to capture the current demand for a given vertical. For example, we could consider the number of queries or the number of clicks within the vertical search interface (if one is available). Moreover, we could consider the number of recently indexed documents in the vertical collection. In this case, the assumption is that vertical supply is correlated with vertical demand.

## 2.5 Query-vertical Features

Query-vertical features measure the relationship between the query and the vertical. In this respect, their values depend precisely on the query-vertical pair under consideration. Query-vertical features can be categorized into *pre-retrieval* and *post-retrieval* features. Pre-retrieval features can be computed without issuing the full query to the vertical in question, while post-retrieval features require issuing the full query to the vertical.

### 2.5.1 Pre-retrieval Query-Vertical Features

Pre-retrieval query-vertical features can be generated without issuing the query to the vertical. For this reason, these features are particularly appropriate for vertical selection versus vertical presentation.

**Co-occurrence features.** Co-occurrence features are motivated by the following intuition. Suppose a user issues the query “buy iphone”.

The term “buy” suggests the user almost certainly wants shopping vertical results. Now, suppose another user issues the query “iphone”. A system might determine that the shopping vertical is still relevant because many queries containing the term “iphone” also contain the term “buy”.

Co-occurrence features consider the extent to which the input query terms frequently appear in other queries containing keywords that are strongly predictive for a particular vertical [Arguello et al., 2011a; Jie et al., 2013; Zhou et al., 2012a]. A system might predict that the shopping vertical is relevant to the query “iphone” because this term has a high degree of co-occurrence with other query terms such as “buy”, “price”, “shop”, and “deal”. Similarly, a system might predict that the local vertical is relevant to the query “pizza” because this term has a high degree of co-occurrence with other query terms such as “local”, “nearby”, “places”, and “restaurants”.

Generating co-occurrence features requires two steps: (1) constructing a list of “trigger” terms for the candidate vertical and (2) measuring the degree of co-occurrence between the input query terms and the candidate vertical’s “trigger” terms. Step 1 is done in advance and only once, while Step 2 is done at query time.

Constructing a list of trigger terms can be done manually or automatically. For a well-understood vertical such as images, one can easily come up with terms that are likely to remain predictive of relevance over time (e.g., “pics”, “images”, “photo”, etc.) An automatic method might consider how frequently the term appears in queries issued directly to the vertical.

Measuring the degree of co-occurrence between the input query terms and the candidate vertical’s trigger terms can be done using any co-occurrence measure. A commonly used one is point-wise mutual information (PMI). Let  $q$  denote the input query and  $T_v$  denote the set of trigger terms associated with vertical  $v$ . The PMI between the  $i^{\text{th}}$  input query term and the  $j^{\text{th}}$  trigger term is given by:

$$\text{PMI}(q_i, T_{v,j}) = \log_2 \left( \frac{P(q_i, T_{v,j})}{P(q_i) \times P(T_{v,j})} \right).$$

The individual and joint term probabilities can be estimated using a

query log. In this case,  $P(q_i, T_{v,j})$  corresponds to the proportion of queries that contain *both* terms,  $P(q_i)$  corresponds to the proportion of queries that contain term  $q_i$  (with or without  $T_{v,j}$ ), and  $P(T_{v,j})$  corresponds to the proportion of queries that contain  $T_{v,j}$  (with or without  $q_i$ ).

Assuming an input query with  $n$  terms and a candidate trigger list of  $m$  terms, the affinity to between query  $q$  and vertical  $v$  can then be measured using the average PMI between all query-/trigger-term pairs:

$$\phi_{q,v}^{\text{co-occur}} = \frac{1}{n \times m} \sum_{i=1}^n \sum_{j=1}^m \text{PMI}(q_i, T_{v,j})$$

In prior work, Arguello et al. [2011a] used a set of manually constructed trigger terms and computed the input query’s affinity to the vertical using the chi-squared statistic and the AOL query-log.

**Vertical query-log features.** In some cases, a vertical may have direct search capabilities. For example, Google users can issue queries directly to the news vertical if they already know they want news results. Query-log features assume that the queries in the vertical query-log exemplify the types of queries for which the vertical is relevant.

One approach is to consider the number of times the query was issued directly to the vertical by users. For example, Diaz [2009] considered the frequency of the query in the last  $k$  queries issued directly to the vertical and in the last  $k$  queries issued directly to the vertical on the previous day.

A different approach is to measure the *similarity* between the input query and the queries in the vertical query-log. A simple approach that has been effective in prior work is to compute the query-generation probability given a language model constructed from the vertical’s query-log [Arguello et al., 2009b; Diaz and Arguello, 2009; Arguello et al., 2010]. Let  $\theta_v^{\text{qlog}}$  denote the query-log language model for vertical  $v$ . The query-generation probability for query  $q$  can be computed as:

$$\phi_{q,v}^{\text{qlog}} = \frac{1}{Z_q} \prod_{w \in q} P(w | \theta_v^{\text{qlog}}). \quad (2.1)$$

The query-generation probability depends on the length of the query. Queries with more query terms will usually have a lower probability



than those with fewer query terms. Machine learned classifiers perform better when features values are comparable across training and test instances. For this reason, it is important to normalize the query-generation probability as follows:

$$\mathcal{Z}_q = \sum_{v \in \mathcal{V}} \left( \prod_{w \in q} P(w | \theta_v^{\text{qlog}}) \right). \quad (2.2)$$

Vertical query-log features have been found to be highly effective. Arguello et al. [2009b] found vertical query-log features to be the best single-evidence predictor. In this case, the authors simply predicted the vertical with the greatest query likelihood (Equation 2.1) if it was above a threshold that was tuned using validation data.

**General query-log features.** Some verticals may not have direct search capabilities and may therefore not have a vertical-specific query-log available. For example, consider the weather vertical in most commercial search engines. The address this issue, Zhou et al. [2012a] proposed the following approach.

Again, let  $T_{v,j}$  denote a set of “trigger” terms associated with vertical  $v$ . For example, for the weather vertical,  $T_{v,j}$  can contain the single term “weather”. Zhou et al. [2012a] used a general query-log find queries related to vertical  $v$  by including all queries with at least one term in  $T_{v,j}$  and all queries with clicks on URLs containing at least one term in  $T_{v,j}$ . Then, the affinity of vertical  $v$  for query  $q$  was measured as:

$$\phi_{q,v}^{\text{gen-qlog}} = \frac{1}{\mathcal{Z}_q} \prod_{w \in q} P(w | \theta_v^{\text{gen-qlog}}),$$

where  $\theta_v^{\text{gen-qlog}}$  denotes a language model generated from all queries related to vertical  $v$ , and  $\mathcal{Z}_q$  is computed as in Equation 2.2.

**Vertical corpus features.** Vertical corpus features aim to estimate the amount of query-related content in the vertical. This is typically done using documents sampled from the vertical.

In cases where vertical documents are only accessible via a search interface, vertical documents can be obtained using query-based sampling [Callan and Connell, 2001b; Shokouhi et al., 2006a; Caverlee et al.,

2006]. The basic idea behind query-based sampling is to gather a sample of vertical documents by issuing random queries to the vertical and downloading the top results. Sampling queries can be generated by randomly selecting terms from the downloaded results [Callan and Connell, 2001a], or by randomly selecting queries for an external query-log [Shokouhi et al., 2006a]. Usually, the same number of documents are sampled from each vertical. Callan and Connell [2001a] found that 300-500 samples per resource is enough for smaller collections, and Shokouhi et al. [2006a] found that larger samples of about 1,000 documents per resource improved resource selection performance for larger collections.

In general, vertical corpus features can be generated in two ways. The simplest approach is to compute the similarity between the query and the set of documents sampled from the vertical. Si et al. [2002] proposed the following “large document” approach for the task of *resource selection* in text-based federated search. Let  $v_s$  denote the set of documents sampled from vertical  $v$  and  $\theta_{v_s}$  denote the language model generated from these samples. The query-sample similarity can be computed as:

$$\phi_{q,v}^{\text{large-doc}} = \frac{1}{\mathcal{Z}_q} \prod_{w \in q} P(w|\theta_{v_s}), \quad (2.3)$$

where normalizer  $\mathcal{Z}_q = \sum_{v \in \mathcal{V}} \left( \prod_{w \in q} P(w|\theta_{v_s}) \right)$ . Again, the raw query likelihood score depends on the query length. Thus, to make these values comparable across queries,  $\mathcal{Z}_q$  normalizes across candidate verticals. This is considered a “large document” approach because the vertical is modeled as a single large document (a virtual concatenation of all documents in  $v_s$ ).

Some vertical documents (e.g., images, videos) may not be inherently associated with lots of text (if any). In such cases, we may be able to use textual metadata information to represent the query and the vertical documents. Duarte Torres et al. [2013] proposed such an approach and used metadata tags from Del.icio.us, a social bookmarking site where users tag webpages using freely chosen index terms.<sup>1</sup> The

---

<sup>1</sup><https://delicious.com/>

input query was represented using the metadata tags associated with the top- $n$  results from an external collection of tagged documents, and each vertical was represented using tags associated with its sampled documents. The query-vertical similarity was then computed based on the probability given to the query-tags from the vertical’s tag-based language model (similar to Equation 2.3). Duarte Torres et al. [2013] found this approach to be highly effective as a single-evidence predictor.

One potential limitation of large document approaches is that they do not directly model the *absolute* number of query-related documents in the vertical. An alternative approach is to use an algorithm like ReDDE [Si and Callan, 2003a], which stands for Relevant Document Distribution Estimation.

The ReDDE algorithm proceeds as follows. First, it combines the samples from each vertical  $v \in \mathcal{V}$  in a *centralized sample index* (CSI). Then, given query  $q$ , it performs a retrieval from the CSI. Let  $\mathcal{R}_{q,csi}^n$  denote the top- $n$  CSI results returned in response to query  $q$ . ReDDE estimates the number of query-related documents in vertical  $v$  as:

$$\phi_{q,v}^{\text{redde}} = \frac{|v|}{|v_s|} \sum_{d \in \mathcal{R}_{q,csi}^n} \mathcal{I}(d \in v_s), \quad (2.4)$$

where  $|v_s|$  denotes the number of documents sampled from vertical  $v$ ,  $|v|$  denotes the total number of documents in  $v$ , and  $\mathcal{I}$  denotes the indicator function, which returns 1 if the argument is true and 0 otherwise.

The intuition behind ReDDE is that each document  $d$  in the top- $n$  CSI results represents exactly  $\frac{|v|}{|v_s|}$  *unseen* documents in the vertical from which document  $d$  originated. Factor  $\frac{|v|}{|v_s|}$  in the equation is responsible to estimating the *absolute* number of query-related documents in the vertical. The ReDDE algorithm, and variants such as Soft.ReDDE [Arguello et al., 2009b], have proven to be important features for the task of vertical selection [Arguello et al., 2009b; Diaz and Arguello, 2009; Arguello et al., 2009a, 2010; Zhou et al., 2012b; Duarte Torres et al., 2013].

Equation 2.4 requires knowing the total number of documents in each vertical. In a cooperative environment, the system might have access to this information. Alternatively, several collection size estima-

tion approaches have been proposed in prior work [Liu et al., 2002; Khelghati et al., 2012; Shokouhi et al., 2006b].

Two simple collection size estimation approaches are: sample-resample [Si and Callan, 2003a] and capture-recapture [Liu et al., 2001].

The *sample-resample* approach works as follows. Let  $df_{i,v}$  denote the number of documents in vertical  $v$  that contain term  $t_i$ , and let  $df_{i,v_s}$  denote the number of documents in  $v_s$  that contain  $t_i$ . If we assume  $v_s$  to be a truly representative sample of  $v$ , then:

$$\frac{df_{i,v}}{|v|} = \frac{df_{i,v_s}}{|v_s|}.$$

If this equality holds true, then the number of documents in  $v$  can be estimated as:

$$|v| = \frac{df_{i,v} \times |v_s|}{df_{i,v_s}}.$$

The *capture-recapture* approach works as follows. Let  $v_{s1}$  and  $v_{s2}$  denote two random samples from vertical  $v$ . Given these two samples, the number of documents in  $v$  can be estimated as:

$$|v| = \frac{|v_{s1}| \times |v_{s2}|}{|v_{s1} \cap v_{s2}|}, \quad (2.5)$$

where the denominator corresponds to the number of documents in common between both samples.

**Pre-retrieval query performance features.** The goal of *query performance prediction* is to automatically estimate a query’s retrieval performance without user feedback or document relevance judgements. Within the context of aggregated search, query performance predictors can be used to favor verticals that are likely to produce an effective retrieval for the given query.

Query performance predictors can be classified into *pre-retrieval* predictors (do not require conducting a full retrieval from the collection) and *post-retrieval* predictors (require conducting one or more full retrievals from the collection).

At their core, query performance predictors assume that well-performing queries are highly topically focused. One simple pre-retrieval approach is to measure the degree of co-occurrence between

the query terms. If the query terms tend to appear together in the same documents, then the query describes a coherent topic with respect to the vertical collection.

Hauff [2010] used the *average* point-wise mutual information (PMI) between query-term pairs. The PMI between query terms  $w_i$  and  $w_j$  is given by:

$$\text{PMI}(w_i, w_j) = \log_2 \left( \frac{P(w_i, w_j)}{P(w_i) \times P(w_j)} \right).$$

In this case,  $P(w_i, w_j)$  corresponds to the proportion of documents in vertical  $v$  that contain *both* terms, while  $P(w_i)$  and  $P(w_j)$  correspond to the proportion of documents in  $v$  that contain term  $w_i$  (with or without  $w_j$ ) and term  $w_j$  (with or without  $w_i$ ), respectively.

**Implicit Feedback Features.** In an operational setting, a system may be able to use previous user interactions to predict whether a vertical is relevant to a query. Implicit feedback features are typically derived from previous clicks on the vertical results, either from the *same* query or *similar* queries. One approach is to simply count the number of times a previous user clicked on results from a particular vertical for the same query.

Another approach is to measure the query-vertical *click-through rate* [Ponnuswami et al., 2011b,a; Wang et al., 2016]. Let  $\mathcal{C}_q^v$  denote the number of times vertical  $v$  was presented for query  $q$  and the user clicked on it, and let  $\mathcal{S}_q^v$  denote the number of times vertical  $v$  was presented for query  $q$  and the user did *not* click on it. The query-vertical click-through rate is given by:

$$\phi_{q,v}^{\text{click}} = \frac{\mathcal{C}_q^v}{\mathcal{C}_q^v + \mathcal{S}_q^v}. \quad (2.6)$$

The same idea can be extended to derive evidence from previous queries that are *similar* to the input query, but not exactly the same. Suppose that the input query is “new york style pizza central park”, which suggests that the system should display results from a local businesses vertical. Furthermore, suppose that this query has not been entered into the system before, but the query “new york pizza central park” has, and has a local vertical click-through rate of about 20%.

One would expect the similar query “new york style pizza central park” to have a similar click-through rate.

Let  $\mathcal{Q}$  denote the set of all previously run queries and let  $\text{sim}(q, q')$  denote a similarity measure between query  $q$  and  $q'$ . The click-through rate for similar queries can be computed as:

$$\phi_{q,v}^{\text{sim-click}} = \frac{1}{\mathcal{Z}} \sum_{q' \in \mathcal{Q}} \text{sim}(q, q') \times \phi_{q,v}^{\text{click}}, \quad (2.7)$$

where normalizing constant  $\mathcal{Z} = \sum_{q' \in \mathcal{Q}} \text{sim}(q, q')$ .

Query similarity can be measured in different ways, for example, based on the query-term overlap, based on the overlap between the top results returned from a particular collection, or based on the similarity between the language models of the top results returned from the vertical or an external collection [Diaz, 2009].

Finally, another approach is to generate a language model from previous queries with clicks on results from the particular vertical. Then, we can measure the query-generation probability given by this language model to the input query [Arguello et al., 2009a, 2011a]. Let  $\theta_v^{\text{click}}$  denote a language model generated from queries with clicks on results from vertical  $v$ . The query generation probability is given by:

$$\phi_{q,v}^{\text{lm-click}} = \frac{1}{\mathcal{Z}} \prod_{w \in q} P(w | \theta_v^{\text{click}}). \quad (2.8)$$

Again, because the query generation probability depends on the query length, it is important to normalize this value by  $\mathcal{Z} = \sum_{v \in \mathcal{V}} \left( \prod_{w \in q} P(w | \theta_v^{\text{click}}) \right)$ .

### 2.5.2 Post-retrieval Query-Vertical Features

Post-retrieval query-vertical features are generated directly from the vertical results in response to the query. Because post-retrieval features are more computationally expensive than pre-retrieval features, these features are typically used for vertical presentation. Post-retrieval query-vertical features measure the *quality* of the vertical results returned in response to the query.

Some of the features used in prior work seem fairly general and are likely to be effective for different verticals. Examples include the

number of results returned by the vertical [Wang et al., 2016], the retrieval score of the top result [König et al., 2009], the average retrieval score of the top- $n$  results [Ponnuswami et al., 2011b], and the average number of query terms appearing in the top- $n$  results [Arguello et al., 2011a]. For the task of vertical presentation, Arguello et al. [2011a] conducted a feature ablation study and found the greatest performance drop from removing features measuring the term overlap between the query and the top vertical results.

Other features are more specific to a particular vertical. For example, Diaz [2009] focused on the task of vertical selection for the news vertical and included features measuring the recency of the top news results: the average publication age of the top- $n$  vertical results and the proportion of most recently indexed news articles in the top- $n$  results. Including such features is motivated by the fact that recency is an important criterion for predicting relevance for news. The same idea could be applied to other verticals. For example, for local the vertical, we could measure the average geographical proximity of the top-ranked locations, and for the video vertical, we could measure the average number of views associated with the top-ranked videos.

**Post-retrieval query performance features.** Post-retrieval query performance predictors estimate a query’s effectiveness from the retrieval itself. For textual verticals, a commonly used query performance predictor is the *Clarity* score [Cronen-Townsend et al., 2002]. Clarity assumes that the top results from an effective query will have a highly topical language model, which will diverge significantly from a general, query-independent language model. To this end, Clarity measures the Kullback-Leibler divergence (or dissimilarity) between the language model of the top-ranked documents and a background language model.

Let  $\theta_{q,v}$  denote a query-biased language model derived from  $v$  and let  $\theta_G$  denote a general, query-agnostic language model. The Clarity score can be computed as:

$$\phi_{q,v}^{\text{clarity}} = \sum_{w \in q} P(w|\theta_{q,v}) \log \left( \frac{P(w|\theta_{q,v_s})}{P(w|\theta_G)} \right). \quad (2.9)$$

Let  $\mathcal{R}_{q,v}^n$  denote the top- $n$  results returned from vertical  $v$  in response

to  $q$ . Language model  $\theta_{q,v}$  can be estimated as:

$$P(w|\theta_{q,v}) = \frac{1}{\mathcal{Z}} \sum_{d \in \mathcal{R}_{q,v}^n} P(w|\theta_d) \times \text{score}(d, q), \quad (2.10)$$

where  $\theta_d$  denotes the language model of document  $d$ ,  $\text{score}(d, q)$  denotes the retrieval score given to document  $d$ , and normalizing factor  $\mathcal{Z} = \sum_{d \in \mathcal{R}_{q,v}^n} \text{score}(d, q)$ . Equation 2.10 essentially averages the top- $n$  document language models in a weighted fashion (weighted by the mass-normalized document retrieval score).

Prior work in aggregated search used the Clarity score as a type of feature, but used retrievals from a collection of documents sampled from the vertical rather than the vertical’s actual retrieval [Arguello et al., 2009b,a; Diaz and Arguello, 2009; Arguello et al., 2010; Zhou et al., 2012b; Duarte Torres et al., 2013].

[König et al., 2009] used a feature similar to *post-retrieval* Clarity (Equation 2.9) for the purpose of vertical selection for the news vertical. Given a query, the authors considered the average text-based similarity between the top-50 results returned by the news vertical search engine. Similar to Clarity, the underlying assumption is that the top results from an effective retrieval should be focused on the same topic.

## 2.6 Summary and Considerations

In this chapter, we reviewed a wide range of features considered in prior vertical selection and presentation research. We considered features generated from the query, the vertical, and the query-vertical pair.

Table 2.1 provides a summary of the different features types discussed in this chapter. We indicate whether the feature is a pre- or post-retrieval features, provide a brief description, and mention any resources or tools required to generate the feature value.

At this point, it is worth highlighting a few important points.

First, not every feature will be available for every vertical. For example, vertical query-log features will not be available for verticals that do not have direct search capabilities. Similarly, vertical corpus features



will not be available for verticals that do have an underlying collection of retrievable items.

Second, not every feature will be equally predictive of relevance for every vertical. This is especially the case for query features (derived from the query and independent of the vertical in question). For example, features that describe the topic of the query will be more effective for verticals that are topically focused (e.g., autos, games, health) than for verticals that cover a wide range of topics (e.g., community Q&A, images, video).

Third, in this chapter, we also characterized features as pre-retrieval and post-retrieval features. As previously mentioned, vertical selection approaches tend to focus on pre-retrieval evidence, while vertical presentation approaches tend to focus on pre- and post-retrieval evidence. That said, in an operational setting, certain queries are likely to be seen over and over again. In this respect, aggregated search systems can also cache post-retrieval feature values to inform future vertical selection decisions for the same query. For example, an end-to-end system can cache the number of image vertical results in response to the query “jaguar” (a type of post-retrieval features) and use this as a vertical selection feature for future impressions of the same query. Moreover, we could imagine *diffusing* post-retrieval feature values across similar queries to be used as vertical selection features, similar to how we diffused click-through rate in Equation 2.8.

Finally, certain verticals are highly dynamic. Consider the task of predicting whether the news vertical is relevant to a query. A query’s newsworthiness is likely to change over time. Given a dynamic environment, certain features may be more effective than others. For example, a vertical feature such as the number of queries recently issued to the news vertical by users may remain predictive over time. Similarly, a query-vertical feature that measures the average age of the top news vertical results may also remain predictive. Other features, for example, query category features that describe the topic of the query may become less effective as the vertical changes. It is important for system designers to consider features that do not grow “stale” over time. Otherwise, the model needs to re-trained periodically to update how it is

combining evidence to make predictions.

**Table 2.1:** Summary of features types discussed in Chapter 2. The *description* provides a brief explanation of the features and the *required resources* indicate the resources or tools required to compute the feature values.

Pre/post	Feature Name	Description	Required Resource
Pre	Query keywords	Precense of certain keywords in the query (e.g., news, pics, shopping)	Manually selected keywords
Pre	Query characteristic	Query length, punctuation, capitalization, named entity types	Named-entity Tagger
Pre	Query class	Navigational, informational, transactional	Query class classifier
Pre	Query category	Query's affinity to pre-defined topical categories	Collection of pre-classified documents or topical classifier
Pre	Query ambiguity	Likelihood of query having ambiguous intent	Query ambiguity classifier
Pre	Vertical click-through rate	Vertical click-through rate for any query	Click data
Pre	Co-occurrence with vertical triggers	Average co-occurrence between query terms and vertical trigger terms	Manually selected trigger terms and query-log
Pre	Vertical query-log	Query likelihood from vertical query-log language model	Vertical query-log
Pre	General query-log	Query likelihood from language model of queries with clicks on vertical documents	Query-log
Pre	Large document query-vertical similarity	Query likelihood from vertical document language model	Sampled vertical documents
Pre	ReDDE	Estimated number of query-related vertical documents	Sampled vertical documents and vertical size estimates
Pre	Query-term co-occurrence in vertical collection	Average point-wise mutual information between query term pairs in vertical collection	Vertical inverted lists
Pre	Query-vertical click-through rate	Clicks divided by views	Click data
Pre	Estimated query-vertical click-through rate	Weighted average click-through rate from similar queries (weighted by similarity)	Click data
Post	Vertical scores	Retrieval scores from top vertical results	Vertical retrieval scores
Post	Clarity	Divergence of top results from a background (general) language model	Top vertical results and background language model
Post	Top vertical results similarity	Average pairwise similarity between top vertical results	Top vertical results

# 3

---

## Approaches for Vertical Selection and Presentation

---

In the previous chapter, we reviewed different sources of evidence that can be used to inform vertical selection and presentation decisions. In this chapter, we review approaches for using these sources of evidence to make predictions. We first review approaches for vertical selection—deciding which verticals to select in response to a query. Then, we review approaches to vertical presentation—deciding where to present each selected verticals relative to the web results and each other.

### 3.1 Vertical Selection

The goal of vertical selection is to decide which verticals to present in response to a query. Vertical selection is essentially a multiclass classification task. Given a query, the system must make a binary decision for each candidate vertical.

#### 3.1.1 Single-evidence Approaches

The simplest vertical selection approaches use a single source of evidence to predict which verticals to select in response to a query. Single-evidence approaches involve two steps: (1) computing the measure for

each candidate vertical and (2) using one or more thresholds to predict which verticals to select and which ones to suppress. One alternative is to use the same threshold for all candidate verticals. This alternative makes sense if we believe the single-evidence measure is *directly* comparable across all candidate verticals. Alternatively, we can use different thresholds for the different candidate verticals.

Several single-evidence approaches have been evaluated in prior work. Arguello et al. [2009b] evaluated a single-evidence predictor derived from vertical query-log data (i.e., from queries issued directly to the vertical by users). This approach measured the query likelihood score given by a language model generated from the vertical query-log (Equation 2.1). The same threshold was applied to make binary predictions for each candidate vertical. The threshold was tuned using training data in the form of a set of queries with relevance judgements for each candidate vertical.

Duarte Torres et al. [2013] evaluated several single-evidence approaches derived from sampled vertical documents. These included the ReDDE score (Equation 2.4), the Clarity score (Equation 2.9) and the query-likelihood score (Equation 2.3). The ReDDE and query-likelihood scores were found to be more effective than the Clarity score, possibly because Clarity scores are not directly comparable across verticals.

Diaz [2009] proposed a single-evidence approach derived from vertical click-through data (previous vertical clicks and skips). Again, let  $C_q^v$  denote the number of times the system selected vertical  $v$  and the user clicked on it, and let  $S_q^v$  denote the number of times the system selected vertical  $v$  and the user did *not* click on it. As described in Equation 2.6, the click-through rate for vertical  $v$  and query  $q$  (denoted as  $\phi_{q,v}^{\text{click}}$ ) is the number of times the vertical was clicked ( $C_q^v$ ) out of the number of times it was displayed on the SERP ( $C_q^v + S_q^v$ ). Finally, the system can decide to select vertical  $v$  in response to  $q$  if the historical click-through rate exceeds a threshold.

The main limitation of this approach is that it requires an exact match between queries. In theory, vertical  $v$  should have a similar click-through rate for similar queries. For example, the news vertical

should have a similar click-through rate for the queries “u.s. presidential elections” and “american presidential elections”. Diaz [2009] suggested *smoothing* the click-through rate for vertical  $v$  and query  $q$  by using the click-through rate for vertical  $v$  and queries similar to  $q$  (Equation 2.8). The basic idea is to share click-through statistics across queries that are likely to have the same intent. As previously mentioned, the similarity between two queries can be computed in different ways, for example, based on the overlap between query terms, based on the overlap between the top results return from an external collection, or based on the similarity between the relevance models associated with both queries (Equation 2.10). Diaz [2009] computed the Bhattacharyya correlation between the relevance models associated with both queries:

$$\text{sim}(q, q') = \sum_w \sqrt{P(w|\theta_q)P(w|\theta_{q'})}.$$

Other query-similarity measures have been proposed in prior work [Metzler et al., 2007; Sahami and Heilman, 2006; Wen et al., 2001].

Single-evidence predictors are simple and intuitive. However, they have to main shortcomings. First, they require that the source of evidence be available for all candidate verticals. Vertical query-log evidence will not be available for verticals that do not have direct search capabilities. Likewise, vertical click-through data will not be available for verticals that are not designed to be clicked. Second, single-evidence predictors rely on a single source of evidence to make predictions. Prior research has found that approaches that combine multiple sources of evidence perform better [Arguello et al., 2009b,a; Hong et al., 2010]. Next, we review approaches that combine multiple sources of evidence for predicting which verticals a relevant to a query.

### 3.1.2 Multiple Evidence Approaches

The most successful approaches for vertical selection using machine learning to combine multiple sources of evidence as input features to a model. All the sources of evidence reviewed in Chapter 2 can be seen as potential features for a vertical selection model.

Combining evidence for vertical selection poses two main challenges. First, certain features may not be available for some verticals. For example, verticals without direct search capabilities will not have vertical query-log features. Second, vertical selection requires learning a *vertical-specific* relationship between certain features and the relevance of a particular vertical. For example, query-category features may be more effective for verticals that are topically focused than for verticals that cover a wide range of topics, such as a community Q&A vertical. In all prior work to date, both of these challenges have been addressed by training *independent* binary classifiers (one per vertical) [Li et al., 2008; König et al., 2009; Diaz and Arguello, 2009; Arguello et al., 2009b, 2010]. In this respect, each classifier can adopt a different feature representation and focus on the features that are uniquely predictive for its corresponding vertical.

Machine learned classifiers use training data to learn a predictive model. In the context of vertical selection, training data is in the form of a set of queries  $\mathcal{Q}_v$  with relevance judgments with respect to vertical  $v$ . The machine learning algorithm uses the training data to learn a predictive relationship between the set of input features and the relevance of the vertical. We can think of a vertical selection model as follows:

$$f(q, v) = g(\phi_{(q,v)}, \theta_v),$$

where  $f(q, v)$  denotes the model’s confidence value that vertical  $v$  is relevant to query  $q$ ,  $\phi_{(q,v)}$  denotes a  $m \times 1$  vector of  $m$  features, and  $\theta_v$  denotes the parameters of the model. Feature vector  $\phi_{(q,v)}$  can include any of the features reviewed in Chapter 2. Function  $g$  and the exact definition of  $\theta_v$  depend on the learning algorithm used.

Prior work on vertical selection has used different machine learned classifiers. One important decision is whether to use a *linear* or *non-linear* classifier.

**Linear Classifiers.** In a linear classifier, each feature contributes to the model’s final prediction, but the model does not exploit *interactions* between features. For example, the model cannot learn that the vertical  $v$  is more likely to be relevant if the value of feature  $i$  is high and the value of feature  $j$  is low (or vice-versa).

As an example of a simple linear classifier, a perceptron classifier would predict that vertical  $v$  is relevant to  $q$  using the following function:

$$f(q, v) = \begin{cases} 1 & \text{if } \phi_{(q,v)} \cdot \theta_v > 0 \\ 0 & \text{otherwise.} \end{cases}$$

In this case,  $\theta_v$  is defined as a  $m \times 1$  vector of feature weights.<sup>1</sup> The algorithm learns parameters  $\theta_v$  such that the classification accuracy in the training set  $\mathcal{Q}_v$  is minimized.

A popular linear classifier used in prior vertical selection work is logistic regression [Li et al., 2008; Diaz, 2009; Arguello et al., 2009b; Diaz and Arguello, 2009]. In the case of logistic regression,  $\phi_{(q,v)}$  is also defined as an  $m \times 1$  vector of feature weights, and  $f(q, v)$  is given by:

$$f(q, v) = \frac{\exp(\phi_{(q,v)} \cdot \theta_v)}{1 + \exp(\phi_{(q,v)} \cdot \theta_v)}.$$

**Non-linear Classifiers.** Other approaches for vertical selection have used non-linear classifiers that are able to exploit feature interactions. König et al. [2009] and Arguello et al. [2010] used the Gradient Boosted Decision Trees (GBTD) algorithm [Friedman, 2002]. The main component of a GBDT model is a regression tree. A regression tree is a simple binary tree. Each internal node corresponds to a feature and a splitting condition which partitions the data. Each terminal node corresponds to a response value, the predicted output value. GBDT combines regression trees in a boosting framework to form a more complex model. During training, each additional regression tree is trained on the residuals of the current prediction.

Non-linear classifiers such as GBDT have advantages and disadvantages. The main advantage is that they can exploit complex interactions between features. This, however, also allows the algorithm to overfit the training data. While it may seem counter-intuitive at first, a more flexible model that is able to perfectly classify the training data may be less able to generalize well to *new* data. Choosing between a linear or non-linear classifier may depend on different factors, such as

---

<sup>1</sup>For simplicity, we are omitting the bias parameter  $b$



the size of the training data and whether we believe that modeling feature interactions is likely to improve vertical selection performance.

### 3.1.3 Adaptive Models

A vertical’s relevance to a query is likely to change over time. This is especially the case for verticals that focus on recent events such as news. For example, the query “boston” may be a newsworthy query during some time periods (when some significant event happens), but not others. Ideally, we would like a vertical selection system that can adapt to changes in users’ demands.

As mentioned in Section 2.6 certain features are useful in building an adaptive model. For example, the input query’s likelihood given by the vertical’s query-log language model is a type of adaptive feature, as long as the language model can be updated to reflect the previous queries *recently* issued to the vertical by users. The feature value should be high when the query “boston” is newsworthy and low otherwise.

Another approach to developing an adaptive solution is to periodically re-train the system using vertical clicks and skips [Diaz, 2009; Ponnuswami et al., 2011b,a; Jie et al., 2013]. This would allow use to exploit features whose values do not change over time (e.g., the query contain the word “boston”). As previously mentioned, a vertical *click* is a click on the vertical results block, and a vertical *skip* is a click on a lower-ranked result, but not the vertical. Clicks can be treated as true positive predictions and skips are treated as false positive predictions. We can use the current model to gather a set of vertical clicks and skips, and then use this data to re-train the model to predict clicks and skips with (hopefully) greater accuracy.

The process outlined above has one main limitation. We can use clicks and skips to reason about the current system’s level of precision. However, because we cannot observe clicks on verticals that are not presented, we cannot reason about the current system’s level of recall. That is, we cannot estimate how often the system *should have* presented the vertical, but did not.

In machine learning, *exploitation* happens when a model outputs the prediction with the greatest confidence value. Conversely, *explo-*

*ration* happens when the model outputs a random prediction in order to obtain feedback. Systems that aim to learn from user feedback must decide how to balance exploitation versus exploration. If we only do exploitation, then the system has no way of improving its level of recall. On the other hand, if we only do exploration (by always making random predictions), then we start deteriorating the user experience. Next, we cover two approaches that can balance exploration versus exploitation.

**$\epsilon$ -greedy.** The simplest approach for balancing exploration and exploitation is referred to as the  $\epsilon$ -greedy approach. In this case, the system outputs its most confident prediction with probability  $1 - \epsilon$ , and outputs a random prediction with probability  $\epsilon$  [Sutton and Barto, 1998]. Parameter  $\epsilon$  controls the level of exploration versus exploitation. A high value (e.g.,  $\epsilon = 0.9$ ) results in a large number of random predictions, while a low value ( $\epsilon = 0.1$ ) results in a small number of random predictions.

**Diaz [2009].** One limitation of the  $\epsilon$ -greedy approach is that level of exploitation versus exploration is stationary. In certain cases, we may want to decrease the level of exploration as we accumulate more user feedback. Diaz [2009] proposed a vertical selection approach that can balance exploration versus exploitation in a more principled way.

Let  $\mathcal{C}_q^v$  and  $\mathcal{S}_q^v$  denote the number of observed clicks and skips for vertical  $v$  and query  $q$ . Suppose we have a vertical selection system for vertical  $v$  and suppose that  $\tilde{p}_q^v$  denote the system's predicted probability that  $v$  is relevant to  $q$ . The system is configured to select  $v$  in response to  $q$  when its prediction confident value exceeds a threshold  $\tau$ .

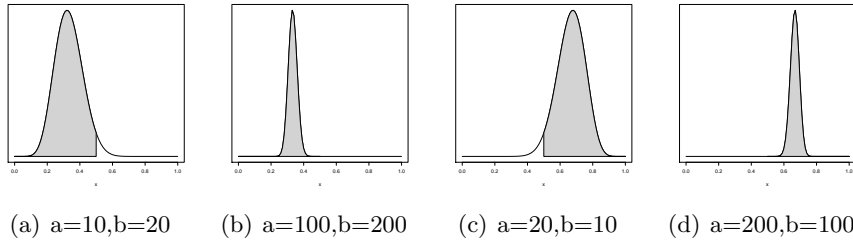
Diaz introduced *exploration* into this framework as follows. Suppose we have a machine learned model that outputs a probability that  $v$  is relevant to  $q$  denoted as  $\pi_q^v$ . Instead of having the final system output  $\tilde{p}_q^v = \pi_q^v$ , we can sample  $\tilde{p}_q^v$  from a Beta distribution defined by the following parameters:

$$\begin{aligned}\tilde{p}_q^v &\sim \text{Beta}(a, b) \\ a &= \mu\pi_q^v + \mathcal{C}_q^v \\ b &= \mu(1 - \pi_q^v) + \mathcal{S}_q^v,\end{aligned}$$

where  $\mu$  is a parameter of the system.

To better understand this approach, let us consider the four different Beta distributions illustrated in Figure 3.1. Figures 3.1(a) and 3.1(b) correspond to cases where  $a < b$ , while Figures 3.1(c) and 3.1(d) correspond to cases where  $a > b$ . Given the above definitions of  $a$  and  $b$ , the value  $a$  is directly proportional to three factors: (1) that machine learned model’s confidence that  $v$  is relevant to  $q$  ( $\pi_q^v$ ), (2) the number of previously observed clicks for the query-vertical pair ( $\mathcal{C}_q^v$ ), and (3) parameter  $\mu$ . Likewise, the value of  $b$  is proportional to: (1) that machine learned model’s confidence that  $v$  is *not* relevant to  $q$  ( $1 - \pi_q^v$ ), (2) the number of previously observed skips for the query-vertical pair ( $\mathcal{C}_q^v$ ), and (3) parameter  $\mu$ .

Suppose we configure the system to display vertical  $v$  in response to  $q$  if  $\tilde{p}_q^v$  is greater than  $\tau = 0.5$ . It is more likely that  $\tilde{p}_q^v < 0.5$  ( $v$  is not selected) in cases where  $a < b$  (Figures 3.1(a) and 3.1(b)), either because  $\pi_q^v$  is low or because  $\mathcal{C}_q^v < \mathcal{S}_q^v$ . Moreover, if we compare the two figures, the likelihood that  $\tilde{p}_q^v < 0.5$  is much *greater* for high values of  $a$  and  $b$  (e.g., more implicit feedback). Conversely, it more likely that  $\tilde{p}_q^v > 0.5$  ( $v$  is selected) in cases where  $a > b$  (Figures 3.1(c) and 3.1(d)), either because  $\pi_q^v$  is high or because  $\mathcal{C}_q^v > \mathcal{S}_q^v$ . Again, the likelihood that  $\tilde{p}_q^v > 0.5$  is much *greater* for high values of  $a$  and  $b$  (e.g., more implicit feedback).



**Figure 3.1:** Beta distributions with different values of parameters  $a$  and  $b$ . Figures 3.1(a) and 3.1(b) represent cases where the system should suppress vertical  $v$  because  $a < b$ . The likelihood that  $\tilde{p}_q^v < 0.5$  (gray area) is much greater when  $a \ll b$ . In a similar fashion, Figures 3.1(c) and 3.1(d) represent cases where the system should select vertical  $v$  because  $a > b$ . The likelihood that  $\tilde{p}_q^v > 0.5$  (gray area) is much greater when  $a \gg b$ .

Given this framework, the variance of  $\tilde{p}_q^v$  depends on four different components: the machine learned model’s confidence that the vertical is relevant ( $\pi_q^v$ ) or not relevant ( $1 - \pi_q^v$ ), the observed number of clicks ( $\mathcal{C}_q^v$ ), the observed number of skips ( $\mathcal{S}_q^v$ ), and the value of parameter  $\mu$ . As all these values increase, the expected value of  $\tilde{p}_q^v$  converges to the mean of the Beta distribution, which in this case corresponds to:

$$\frac{\mathcal{C}_q^v + \mu\pi_q^v}{\mathcal{C}_q^v + \mathcal{S}_q^v + \mu}.$$

This framework has several nice properties. First, the level of exploration is greater in cases where the machine learned model is *less* confident about the vertical’s relevance or non-relevance for the query. Second, parameter  $\mu$  controls the amount of exploration as well as the amount of confidence given to the machine learned model in light of previously observed clicks and skips. Finally, as the system observes more clicks and skips,  $\tilde{p}_q^v$  approximates the click-through rate:

$$\frac{\mathcal{C}_q^v}{\mathcal{C}_q^v + \mathcal{S}_q^v}.$$

### 3.2 Vertical Presentation

The goal of vertical presentation is to decide where to present the selected verticals relative to the web results and each other. In general, vertical presentation is a more difficult task than vertical selection, for several reasons. First, if we assume graded relevance, the goal of the system is to present the more relevant vertical or web results in a more salient way. In practice, this translates to presenting the most relevant results higher on the SERP. Thus, the system must predict the *degree* of relevance of a vertical to a query. Second, the system must consider different factors in deciding where to present each selected vertical. At the very least, the system must consider the query’s vertical intent as well as the quality of the results returned by the vertical. For example, while the query “buy iphone” clearly has shopping vertical intent, a system may decide to present the shopping results lower on the SERP if the results appear to be poor. In fact, in some cases, a

system may even have the option of not displaying a previously selected vertical in light of post-retrieval evidence. Finally, vertical presentation systems have to deal with the fact that different verticals are associated with different levels of visual salience. So, for example, displaying non-relevant images in the middle of the SERP may have a more negative effect than displaying non-relevant news results in the same position.

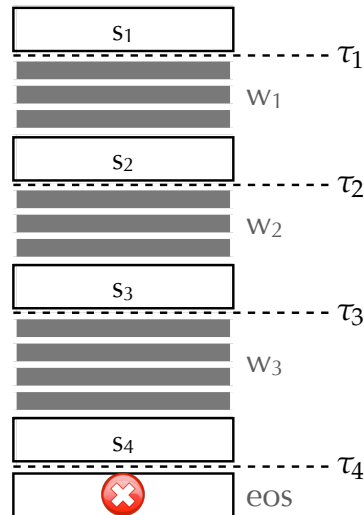
Previously proposed approaches to vertical presentation can be classified into two different types: *pointwise* and *pairwise* approaches. In both cases, each selected vertical  $v$  corresponds to a *block*—a sequence of  $t_v$  results that must be presented together on the SERP. In current systems, certain verticals (e.g., news) organize the results vertically within the block, while other verticals (e.g., images) organize the results horizontally.

### 3.2.1 Pointwise Approaches

Pointwise approaches *directly* predict the degree of relevance of each vertical block to a query. In this respect, pointwise approaches for vertical presentation can be very similar to vertical selection approaches. We can train independent, *vertical-specific* classifiers to predict the degree of relevance of a vertical to a query and use the prediction confidence values the different classifiers to decide where to present each selected vertical. By training independent classifiers we can have each classifier use a different feature representation and learn a vertical-specific relationship between feature values and the relevance of the corresponding vertical.

Several pointwise approaches investigated in prior work assume that vertical blocks can only be presented in specific *slots* within the web results [Arguello et al., 2011a; Ponnuswami et al., 2011b,a]. This idea is illustrated in Figure 3.2. In this example, vertical blocks can only be presented above the first web result (slot  $s_1$ ), between the third and fourth web result (slot  $s_2$ ), between the sixth and seventh web result (slot  $s_3$ ), and below the tenth web result (slot  $s_4$ ).

Pointwise approaches investigated in prior work trained independent, *vertical-specific* classifiers to predict the degree of relevance of a vertical to a query [Arguello et al., 2011a; Ponnuswami et al., 2011b,a].



**Figure 3.2:** Some pointwise approaches to vertical presentation assume that vertical results can only be slotted in certain position on the SERP ( $s_1 - s_2$ ). EOS denotes the end of SERP.

Arguello et al. [2011a] used training data produced by human annotators. Each classifier was trained to predict whether the vertical should be presented on the SERP or not. Ponnuswami et al. [2011b,a] used training data derived from previous vertical clicks and skips. Each classifier was trained to predict vertical clicks and skips when the vertical was displayed above the first web result. In all three studies, vertical blocks were positioned on the SERP using different thresholds, denoted as  $\tau_1 - \tau_4$  in Figure 3.2, where  $\tau_1 > \tau_2 > \tau_3 > \tau_4$ . Each vertical block was presented in the top-ranked slot  $s_i$  as long as the vertical prediction confidence value exceeded threshold  $\tau_i$ . Vertical blocks within the same slot were ordered in descending order of prediction confidence value.

In terms of features, Arguello et al. [2011a] used some of the pre- and post-retrieval features described in Chapter 2. Interestingly, Ponnuswami et al. [2011b,a] also included features derived from the top-10 web results on the SERP. Displaying vertical results in the top slots *displaces* more of the web results below the fold. To model the vertical's

relevance relative to the web results, the authors considered features derived from the web results, including the retrieval score of the top web result, the average retrieval scores across the top-10 web results, and the query/web-result click-through rate for each of the top-10 web results.

Jie et al. [2013] proposed a pointwise approach for vertical presentation with fewer presentation constraints—vertical blocks could be presented above the web results, below the web results, and in between any two web results. Training data for each vertical was generated using previous clicks and skips when the vertical was displayed anywhere on the SERP. Verticals were slotted on the page using the following approach. In addition to training vertical specific classifiers, the authors trained a model to predict clicks and skips for web results using different features, including the web result rank, retrieval score, and the query category. At test time, the slotting mechanism proceeded down the web results (from top-to-bottom) and slotted vertical  $v$  above web result  $w_i$  if the predicted click probability for  $v$  was greater than the predicted click probability for  $w_i$ .

Pointwise approaches have the advantage that they are simple and intuitive. Vertical-specific classifiers can use different feature representations and learn to predict relevance using evidence that is uniquely predictive for the corresponding vertical. The main challenge is that the prediction confidence values from independent classifiers are not always *directly* comparable. Pairwise approach address this issue by learning to predict the *relative* relevance between pairs of blocks.

### 3.2.2 Pairwise Approaches

Pairwise approaches learn to predict the *relative* relevance between candidate block-pairs to be displayed on the SERP. Arguello et al. [2011a] proposed a pairwise approach that proceeds as follows. Let  $\mathcal{B}_q$  denote the set of vertical and web blocks to be displayed in response to query  $q$ . Furthermore, let us assume that vertical blocks can only be displayed in specific slots on the SERP. If we assume the four slots depicted in Figure 3.2, then  $\mathcal{B}_q$  would include one block for each vertical selected in response to query  $q$  and three web blocks that are always

displayed (denoted as  $w_1$ ,  $w_2$ , and  $w_3$ ).

The approach from Arguello et al. [2011a] was to train one binary classifier per block-type-pair. Here, a block *type* refers to the search system that produced the block (i.e., the particular vertical or the web search engine). If we assume  $n$  different verticals, then we would train  $\binom{n}{2} + n$  different classifiers. The first term corresponds to those classifiers trained to predict the relative relevance between blocks from two different verticals, while the second term corresponds to those classifiers trained to predict the relative relevance between blocks from a particular vertical and a web block. Each classifier can use its own feature representation, which can be thought of as the concatenation between two feature vectors: those features thought to be predictive for the first block type and those feature thought to be predictive for the second block type. Arguello et al. [2011a] generated training data from human-produce preference judgements on block-type pairs for a set of queries.

At test time, the approach Arguello et al. [2011a] proceeds in two steps: (1) predict the relative relevance between all candidate block-pairs  $(b_i, b_j) \in \mathcal{B}_q$  and (2) use the predicted pairwise preferences to generate an aggregated SERP. The first step is just a matter of producing preference predictions using the appropriate binary classifiers. The second step is more complicated. Arguello et al. [2011a] used the Schulze Voting Method to derive a block-ranking from the predicted preferences [Schulze, 2011].

The general idea behind the Schulze voting method is the following. Let  $\pi_q(b_i, b_j)$  denote the strength with which block  $b_i$  is preferred over block  $b_j$  (in this case, the output of pairwise classifier associated with block-types of  $b_i$  and  $b_j$ ). We say that  $b_i$  directly defeats  $b_j$  if  $\pi(b_i, b_j) > \pi(b_j, b_i)$ . A *beatpath* from  $b_i$  to  $b_j$  is defined as a direct or indirect defeat from  $b_i$  to  $b_j$ . An *indirect* beatpath from  $b_i$  to  $b_j$  is a sequence of direct defeats from  $b_i$  to  $b_j$ . For example, if  $b_i$  directly defeats  $b_k$  and  $b_k$  directly defeats  $b_j$ , then this is an *indirect* beatpath from  $b_i$  to  $b_j$ . The strength of an indirect beatpath corresponds to the strength associated with the weakest direct defeat in the beatpath. Finally, we say that  $b_i$  defeats  $b_j$  if the strongest (direct or indirect) beatpath from  $b_i$  to  $b_j$  is stronger



than the one from  $b_j$  to  $b_i$ . Finally, blocks are ranked by their number of defeats.<sup>2</sup>

Pairwise approaches such as the one described above also have advantages and disadvantages. As with pointwise methods, each independent classifier can use its own feature representation and focus on the evidence that is uniquely predictive for the block-type-pair in question. Moreover, there are principled ways of combining a set of pairwise preferences into a ranking of items. The main disadvantage is that it requires training a large number of pairwise classifiers, which is cumbersome if we have a large number of verticals. Next, we review learning-to-rank approaches, which require training only a *single* model.

### 3.2.3 Learning-To-Rank Approaches

In machine learning, learning-to-rank (LTR) algorithms learn to order items as a function of a set of features. In the context of information retrieval, LTR algorithms have been used mostly for ranking documents in response to a query. In this case, predictive features are typically generated from the query-document pair and the document (independent of the query).

Existing LTR methods can be classified into three types. *Point-wise* methods (e.g., Gradient Boosted Decision Trees [Friedman, 2002]) learn to predict a document’s relevance grade independent of other documents. *Pair-wise* methods (e.g., RankSVM [Joachims, 2002]) learn to predict whether one document is more relevant than another. *List-wise* methods (e.g., AdaRank [Xu and Li, 2007]) directly optimize an IR evaluation measure such as NDCG, which considers the quality of the ranking as a whole. LTR methods have also been applied to other IR tasks such as ranking query suggestions [Santos et al., 2013], ranking query autocomplete query candidates [Shokouhi, 2013], and ranking related news articles for an input article [Lv et al., 2011].

Using LTR for vertical presentation poses two main challenges. First, LTR approaches require a common feature representation. In the

---

<sup>2</sup>If we assume that web blocks must be presented in their original order, then we can set  $\pi_q(w_i, w_j) = \infty, \forall i < j$ .

context of vertical presentation, certain blocks may not have certain features available. For example, certain verticals may not provide retrieval scores. If we want use vertical retrieval scores to generate post-retrieval features, then these features will not be available for some block types. Second, some LTR approaches assume a *consistent* predictive relationship between features and the relevance of an item. Specifically, this is the true for *linear* LTR models. So, for example, in the context ad-hoc document retrieval, a linear LTR model may assume that the BM25 score between a query and a document is positively predictive of relevance for all documents. In the context of vertical presentation, certain features (e.g., the presence of the query term “news”) will not be predictive in the same direction for different block types.

In general, there are at least three ways of addressing these two challenges.

The first solution is to only use features that are available for all block-types and are expected to be *equally* predictive of relevance across all block-types. Query-vertical and vertical features are good candidate features that meet this criterion. For example, the historic query-vertical click-through rate is likely to be predictive in the same directly for different block-types. The main limitation behind this approach is it ignores evidence that is unique predictive for a particular block-type.

The second and third solutions are more complex. Suppose that each web and vertical block is represented by a feature vector of size  $m$ . Furthermore, suppose that we have  $n$  different block-types. The second alternative is to introduce  $n$  binary features (also referred to as *indicator* features) into the feature representation. The resulting feature representation would be of size  $m + n$ . The goal of these indicator features is to identify the *type* associated with a candidate block. For example, one indicator feature could represent the image vertical, another could represent the news vertical, another could represent the web search engine, and so on. For each instance, one indicator feature is set to '1' (the one corresponding to the block-type of the instance) and the rest are set to '0'. Given this augmented feature representation, we can use a non-linear LTR algorithm such as GBDT and hope that the algorithm learns to exploit useful interactions between indi-

cator and non-indicator features. For example, the model could learn that query term “news” is positive evidence if the news vertical indicator feature is ‘1’, but not if it is ‘0’. An obvious risk with this second approach is that the learning algorithm may not discover these useful feature interactions.

The third solution is to *explicitly* include interactions between all indicator and all non-indicator features in the feature representation. In this case, each interaction feature represents the product between an indicator feature and a non-indicator feature. The resulting feature representation would be of size  $m \times n$ . Note that most feature values would equal zero for each candidate block. In fact, for each instance the number of zero-ed features would be  $m \times n - m$ , and only  $m$  features per instance would not necessarily equal zero.

Prior work explored different LTR approaches for vertical presentation [Arguello and Capra, 2012]. Here, the vertical presentation task was cast as a *block-ranking* task—ranking vertical and web blocks in response to a query. Results found that the third approach outline above outperformed the first. The second approach was not tested.

### 3.3 Summary

In this chapter, we reviewed different approaches for combining sources of evidence to make vertical selection and presentation decisions. Most vertical selection approaches use independent binary classifiers (one per vertical). In this way, each classifier can use its own feature representation and focus on the evidence that is uniquely predictive for the corresponding vertical.

Vertical presentation solutions are more varied. Pointwise approaches are similar to vertical selection approaches. The main difference is that they can harness post-retrieval evidence and must make decisions about where to slot each selected vertical in the web results. Pairwise methods learn to predict the relative relevance between vertical and web block pairs, and can use a voting approach to construct the final SERP. Finally, learning-to-rank (LTR) methods learn a single model to rank blocks in response to a query. LTR methods may require

augmenting the feature representation so that the model can exploit evidence that is not consistently predictive across block-types.

# 4

---

## Evaluation

---

Evaluation is critical to all subfields of information retrieval, and the same is true for aggregated search. Evaluation facilitates the objective comparison between different sources of evidence for predicting vertical relevance, different algorithms for combining sources of evidence, and different parameter configurations for a particular system.

As previously mentioned, aggregated search involves two sub-tasks: (1) predicting which verticals to display in response to a query (vertical selection) and (2) predicting where in the web results to display each selected vertical (vertical presentation). Vertical selection involves predicting which verticals to present and which verticals to suppress. Vertical presentation involves resolving contention between the different selected verticals and presenting the most relevant verticals in a more salient way. In practice, this typically means presenting the most relevant verticals higher on the aggregated SERP.

In some cases, we may want to evaluate the vertical selection component in isolation. In this case, the evaluation focuses on the system's ability to predict which verticals are relevant to a query and which verticals are not. In Section 4.1, we review evaluation methods used in prior vertical selection research. In other cases, we may want to evaluate

the end-to-end system, which includes vertical selection and presentation. In Section 4.2, we review methods for evaluating the end-to-end output of an aggregated search system.

## 4.1 Vertical Selection Evaluation

The goal of vertical selection is to predict which verticals are relevant to a query. Given a query, the vertical selection system makes a binary prediction for each candidate vertical: to select the vertical or not. In general, a good vertical selection system is one that correctly selects all the relevant verticals and correctly suppresses all the non-relevant ones.

### 4.1.1 Vertical Selection Evaluation Metrics

We review vertical selection evaluation metrics using the following notation. Let  $\mathcal{Q}$  denote the set of evaluation queries and  $\mathcal{V}$  denote the set of candidate verticals. As is often the case in IR, we typically care about *average* performance, either by averaging across queries in  $\mathcal{Q}$  or verticals in  $\mathcal{V}$ . To facilitate both options, let  $\mathcal{Q}_v$  denote the set of evaluation queries for which vertical  $v$  is relevant and  $\tilde{\mathcal{Q}}_v$  denote the set of evaluation queries for which the system *predicts*  $v$  to be relevant. Likewise, let  $\mathcal{V}_q$  denote the set of verticals that are relevant to query  $q$  and  $\tilde{\mathcal{V}}_q$  denote the set of verticals the system *predicts* are relevant to  $q$ .

**Accuracy.** Fundamentally, vertical selection is a multiclass classification problem. Thus, all metrics that are relevant to multiclass classification also apply to vertical selection.

A widely used evaluation metric in multiclass classification is *accuracy*. There are two types of correct predictions that a vertical selection system can make in response to a query. The system can either correctly predict that a particular vertical is relevant (a *true positive* prediction) or correctly predict that a particular vertical is not relevant (a *true negative* prediction). In the context of vertical selection, accuracy measures the percentage of true positive and true negative predictions

across all queries and verticals:

$$\mathcal{A} = \frac{1}{|\mathcal{Q}| \times |\mathcal{V}|} \sum_{q \in \mathcal{Q}} \sum_{v \in \mathcal{V}} \mathcal{I}(v \in \mathcal{V}_q \wedge v \in \tilde{\mathcal{V}}_q) \vee \mathcal{I}(v \notin \mathcal{V}_q \wedge v \notin \tilde{\mathcal{V}}_q).$$

The first component denotes a true positive prediction with respect to query  $q$  and vertical  $v$ , and the second component denotes a true negative prediction with respect to  $q$  and  $v$ .

Accuracy has two main drawbacks. The first drawback is that accuracy, by design, masks the types of errors being made. In some cases, we may want to know whether the system is making more false positive or false negative vertical relevance predictions. The second drawback is that accuracy values may be difficult to interpret. To illustrate, a system that selects every vertical for every query (or suppresses every vertical for every query) will almost certainly have an accuracy value greater than zero. In fact, given a query, only a *few* verticals (if any) are likely to be relevant. Therefore, a system that suppresses every vertical for every query is likely to achieve a high accuracy value.

**Precision, Recall, and F-measure.** Metrics such as *precision* and *recall* can help address both of these issues associated with accuracy. Precision and recall can be measured by macro-averaging across queries or across verticals. Precision and recall macro-averaged across queries is given by:

$$\mathcal{P}_{\mathcal{Q}} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{|\mathcal{V}_q \cap \tilde{\mathcal{V}}_q|}{|\tilde{\mathcal{V}}_q|}$$

$$\mathcal{R}_{\mathcal{Q}} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{|\mathcal{V}_q \cap \tilde{\mathcal{V}}_q|}{|\mathcal{V}_q|}$$

In this case, for a given query  $q$ , precision measures the system's ability to reject the non-relevant verticals from the predicted set, while recall measures the system's ability to include the relevant verticals in the predicted set.

Precision and recall macro-averaged across verticals is given by:

$$\mathcal{P}_{\mathcal{V}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{|\mathcal{Q}_v \cap \tilde{\mathcal{Q}}_v|}{|\tilde{\mathcal{Q}}_v|}$$

$$\mathcal{R}_{\mathcal{V}} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \frac{|\mathcal{Q}_v \cap \tilde{\mathcal{Q}}_v|}{|\mathcal{Q}_v|}$$

In this case, for a given vertical  $v$ , precision measures the system’s ability to suppress the vertical when it is not relevant, while recall measures the system’s ability to select the vertical when it is relevant. Macro-averaging performance across queries emphasizes robustness across queries, while macro-averaging performance across verticals emphasizes robustness across verticals.

Independent of whether we compute precision and recall macro-averaged across queries or verticals, in some cases we may want a single metric that measures the balance between precision and recall [Zhou et al., 2012a]. In this case, the *f-measure* is equivalent to the harmonic mean of precision and recall:

$$\mathcal{F}_* = \frac{2 \times \mathcal{P}_* \times \mathcal{R}_*}{\mathcal{P}_* + \mathcal{R}_*}$$

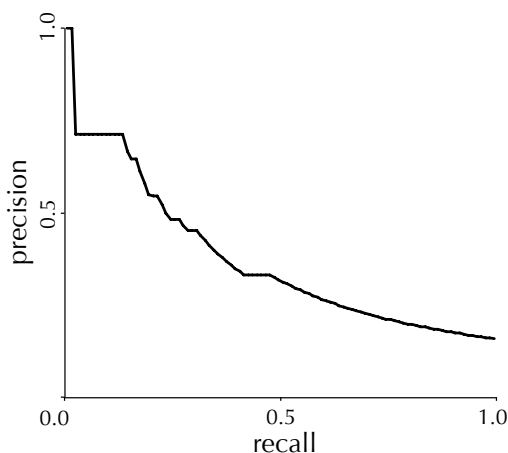
**Precision-Recall Curves.** Machine learned classifiers usually output a prediction confidence value in addition to a binary decision. In such cases, one can introduce a threshold parameter  $\tau$ . The basic idea is to have the system select vertical  $v$  in response to query  $q$  *only* if the classifier’s prediction confidence value is above  $\tau$ . Parameter  $\tau$  can be tuned to favor precision over recall or vice-versa. If we assume that the classifier’s confidence values are in the  $[0,1]$  range, with higher values indicating a higher confidence that  $v$  is relevant to  $q$ , then we can set  $\tau$  to a high value ( $\tau = 0.90$ ) to favor precision over recall, or we can set  $\tau$  to a low value ( $\tau = 0.10$ ) to favor recall over precision.

Parameter  $\tau$  can be introduced into the vertical selection evaluation process in two ways. One alternative is to tune parameter  $\tau$  using a *validation set*. This process involves three steps: (1) evaluating different values of  $\tau$  using a validation set, (2) selecting the value of  $\tau$  with the best performance in terms of some metric of choice (e.g., f-measure



macro-averaged across queries), and (3) evaluating the system with the best parameter value on a held out *test set*.

A second alternative is to measure precision and recall for different values of  $\tau$  (say,  $\tau = 0.0, 0.1, 0.2, \dots, 1.0$ ) and report different precision-recall operating points. A *precision-recall curve* (or *PR curve*) is a graph that visualizes precision (in the  $y$ -axis) as a function of recall (in the  $x$ -axis). A PR curve provides a more complete picture of a system's trade-off between precision and recall. Given two competing systems, the ideal case is to have one system achieve higher values of precision for *all* values of recall. In this case, it is unquestionable that the system with the greater area under the PR-curve is better. Alternatively, we can focus on the precision values associated with the level of recall we think is more important to users. If we think that users typically want to see every vertical that is relevant, then we can focus on precision values associated with high levels of recall. On the other hand, if we think that users *do not* want to see every vertical that is relevant, then we can focus on precision values associated with low levels of recall. Figure 4.2 illustrates an example precision-recall curve.



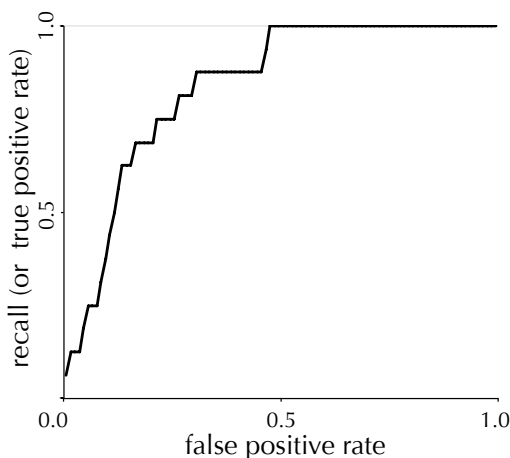
**Figure 4.1:** A precision-recall- or PR-curve shows precision as a function of recall

Prior research in vertical selection has evaluated using PR-curves constructed in two different ways. One approach is to calculate pre-

cision and recall macro-averaged across queries [Duarte Torres et al., 2013]. In this case, the evaluation focuses on the system’s ability to rank verticals in descending order of their relevance to a query. This is analogous to how we construct PR-curves in document ranking. The second approach is to calculate precision and recall for each vertical independently [Li et al., 2008; König et al., 2009]. In this case, the evaluation focuses on the system’s ability to rank *queries* in descending order of the vertical’s relevance to a query.

**Receiver Operating Characteristic (ROC) Curves.** Prior work in aggregated search has not used ROC curves for evaluation. However, an ROC curve conveys similar information as a PR curve. An ROC curve plots recall (in the  $y$ -axis) as a function of the false positive rate (in the  $x$ -axis). Suppose we wanted to use an ROC curve to evaluate a vertical selection model for vertical  $v$  given a set of evaluation queries. Again, let  $\mathcal{Q}$  denote the set of evaluation queries and  $\mathcal{Q}_v$  denote the subset of  $\mathcal{Q}$  for which  $v$  is relevant. An ROC curve is constructed by completing the following steps: (1) Rank all queries in  $\mathcal{Q}$  in descending order of prediction confidence value that  $v$  is relevant; (2) Proceed down this ranking, and at each rank  $k$ , plot recall (or the true positive rate): (i.e., the % of  $\mathcal{Q}_v$  within the top- $k$ ) in the  $y$ -axis and the false positive rate (i.e., the % of  $\mathcal{Q} - \mathcal{Q}_v$  within the top- $k$ ) in the  $x$ -axis. The best ROC curve is one with an area of 1.0—all queries for which  $v$  is relevant ( $\mathcal{Q}_v$ ) are ranked above those for which  $v$  is not relevant. Figure 4.2 illustrates an example ROC curve.

**Rank-based Metrics.** Prior work has also evaluated vertical selection by directly measuring the system’s ability to rank the candidate verticals in descending order of their relevance to the query. This was the evaluation methodology adopted in the TREC Federated Search Track, which ran on 2013 and 2014 [Demeester et al., 2013, 2014]. For a given query, each vertical was assigned a relevance grade proportional to number of relevant documents in its top-10 results. Participating systems were then asked to produce a ranking of verticals in response to each evaluation query, and systems were evaluated using NDCG@20 (Normalized Discounted Cumulative Gain) [Järvelin and Kekäläinen, 2002].



**Figure 4.2:** An ROC curve shows recall (or true positive rate) performance as a function of the false positive rate.

$\text{NDCG}@k$  is computed as follows. Let  $r(i)$  denote the relevance grade associated with the vertical at rank  $i$ .  $\text{DCG}@k$  is given by:

$$\text{DCG}@k = \sum_{i=1}^k \frac{2^{r(i)} - 1}{\log_2(i + 1)}.$$

$\text{DCG}@k$  is not in the  $[0,1]$  range. Thus,  $\text{NDCG}@k$  is computed by dividing  $\text{DCG}@k$  by the *best possible* (or ideal)  $\text{DCG}@k$  value for the given query ( $\text{IDCG}@k$ ):

$$\text{NDCG}@k = \frac{\text{DCG}@k}{\text{IDCG}@k}.$$

$\text{IDCG}@k$  can be computed by simply computing  $\text{DCG}@k$  for a ranking of verticals in descending order of relevance grade (the ideal ranking for the query).

#### 4.1.2 Vertical Relevance Judgements

Vertical selection evaluation requires knowing which verticals are relevant to each query in the evaluation set. Ultimately, we need either binary or graded relevance assessments for each query-vertical pair.

Prior efforts in gathering vertical relevance assessments vary across five different dimensions.

**Anchored vs. Unanchored Assessments.** One might view the vertical selection task as that of selecting a particular vertical *only* if it adds value or complements the core web results, which are always included on the SERP. Prior work investigated the option of asking assessors to judge the relevance of each candidate vertical relative to the web results. In this respect, the vertical relevance assessments are *anchored* on the web results. Zhou et al. [2012a] anchored their vertical relevance judgements *implicitly* by asking assessors whether a candidate vertical might complement the web results for the given search task. In this case, the assessors did not actually see the vertical and web results.

In a follow-up study, Zhou et al. [2013a] investigated the differences between relevance judgements anchored implicitly and *explicitly*, by displaying the actual vertical and web results side-by-side. Results found two interesting trends. First, agreement between assessors was slightly *lower* when the vertical relevance assessments were anchored explicitly. One possible explanation is that seeing the actual vertical and web results caused assessors to take more factors into consideration when making a judgement (e.g., the vertical’s relevance to the search task, the relevance of the top vertical results, and the aesthetics of the vertical results). Second, the authors experimented with anchoring the vertical relevance judgements with web results of different quality (web results 1-3, 4-6, and 7-10). Interestingly, there were no significant differences between the assessors’ judgements.

**Expert Assessors vs. Crowdsourced Assessors.** Several studies used trained assessors who were employees of a commercial search engine company and had expert knowledge of the different candidate verticals [Arguello et al., 2009b, 2010]. Assessors were given a random sample of queries that were issued by real users to the search engine’s main portal and were asked to select which candidate verticals were most likely to be relevant to the user.

Prior work has also gathered vertical relevance judgements from crowdsourced workers [Zhou et al., 2013a]. Crowdsourced workers may have less training than expert assessors. Thus, a commonly used strat-

egy is to gather *redundant* assessments from multiple crowdsourced workers and to derive final assessments using a majority vote [Arguello et al., 2011b; Zhou et al., 2012c]. A study from the early days of Amazon Mechanical Turk found that combining redundant crowdsourced assessments using a majority vote can approximate an expert’s assessments for different tasks in the field of natural language processing [Snow et al., 2008]. In the context of aggregated search, the level of agreement between crowdsourced and expert assessors has not yet been investigated.

**Query vs. Query + Narrative.** Another differentiating factor is whether the assessors were given a query and a description of the user’s information need [Zhou et al., 2014] or whether the assessors were *only* given a query [Arguello et al., 2009b, 2010]. This distinction may seem subtle at first. However, assessments gathered without providing a description are likely to reflect the search intents of *different* users. For example, given only the query “flowers”, an assessor may determine that the images, local, shopping and Q&A verticals are likely to be relevant. In this case, the assessor might be thinking about the different possible intents from different users who might enter this query. If a narrative had been provided along with the query (e.g., “The user is wants to buy flowers or have flowers delivered for a friend.”), an assessor might select a narrower set of relevant verticals that are relevant to the specific information need (e.g., the local and shopping verticals).

Relevance judgements created using only the query (and no narrative) may be more appropriate for evaluating systems based on their ability to diversify the aggregated search results in order to satisfy the search intents from different users.

**Vertical-level vs. Document-level Assessments.** In general, there are two ways to derive vertical relevance judgements from assessors. One alternative is to ask assessors which verticals are relevant to the query. In this case, assessors make judgements based on the user’s intent and their own expectations about the results a vertical might return in response to the query. A second alternative is to ask assessors to judge the top results from each vertical, and to aggregated these document-level relevance judgements up to the vertical level.

Zhou et al. [2014] present a user study that compared these two different ways of gathering vertical relevance judgements. Results found a high level of agreement between vertical relevance judgements derived in both ways. Thus, if the ultimate goal is to gather vertical-level relevance judgements for a set of queries, it may not be necessary to gather relevance judgements on the top results from each vertical.

**Explicit vs. Implicit Assessments.** An alternative to gathering explicit relevance judgements from assessors, is to use implicit feedback. In the context of vertical selection, implicit feedback comes in the form of vertical clicks and skips. A vertical *click* is defined as a click anywhere on the vertical block. On the other hand, a vertical *skip* is defined as a click on a lower-ranked result, but not the vertical. A vertical click signals that the vertical was relevant to the user and a skip signals that it was not relevant.

Diaz [2009] and König et al. [2009] evaluated a news vertical selection system using implicit feedback. Clicks and skips were generated by *always* displaying the news vertical for a small percentage of query traffic. The news vertical was always displayed in the top position (above the first web result). In both studies, model development and evaluation was done *retrospectively*. In other words, the system logged all queries, all vertical clicks and skips, and cached the top web and news vertical results for feature generation. Then, using these resources, the authors evaluated vertical selection systems based on their ability to select the news vertical in cases where it was clicked and suppress the new vertical in cases where it was skipped. Because the news vertical was always displayed for this percentage of query traffic, it was possible to compute click-based precision and recall.

Clicks only suggest *perceived* relevance. In other words, the landing page of a clicked vertical result may not actually provide useful information. To address this issue, a common approach is to also consider dwell time (the period of inactivity between the click and the next SERP event). Prior work has also considered inferring search result relevance using mouse movement information on the landing page [Lagun et al., 2014a]. Mouse movements on the landing page can be captured user a browser toolbar. Lagun et al. [2014b] describe a method for discover-

ing common mouse movement patterns (or *motifs*), which can then be used as features for a predictive model of landing page relevance. Prior work has not considered whether predictive motifs on landing pages from one type of vertical (i.e., images) are also predictive for landing pages from another source (i.e., web pages). Otherwise, one might need to learn different models for different sources.

## 4.2 End-to-end Evaluation

The goal of end-to-end evaluation is to evaluate the final output from an aggregated search system. End-to-end evaluation is less straightforward than vertical selection evaluation. For example, consider the aggregated SERP shown in Figure 1.1. The basic question in end-to-end evaluation is: How good is this particular presentation of results? What if we omitted the news vertical results? How much would this impact the user’s experience? What about a more subtle change, such as swapping the image and video vertical results? Would users even notice?

In this section, we review different methods for end-to-end evaluation. We focus on whole-page evaluation methods, test collection evaluation methods, and on-line evaluation methods.

### 4.2.1 Whole-page Evaluation

Possibly the most straightforward way to determine the quality of an aggregated SERP is to simply ask users. This is the basic intuition behind *whole-page evaluation*. Bailey et al. [2010a,b] proposed the Student Assignment Satisfaction Index (SASI) approach. The general idea is to assess the quality of an aggregated SERP similarly to how a teacher might evaluate a student’s assignment—by determining the extent to which the aggregated SERP satisfies a number of predefined criteria. The SASI interface presents the aggregated SERP to an assessor and asks the assessor to judge individual components on the SERP (e.g., top vertical results, web results 1-3, middle vertical results, web results, 4-10, etc.). Additionally, the assessor is asked to judge the whole SERP along different dimensions (e.g., authority, freshness, diversity, caption quality, overall quality).

The SASI approach has two main advantages. Typically, quality assessments on search results are made out of context. In other words, the relevance of a particular result is independent of the relevance of another. Quality assessments made out of context do not consider factors such as redundancy—a relevant result may be less useful to a user if it contains information that is redundant with a higher-ranked result. In the SASI approach, each component on the SERP is judged within the context of all the other components. So, for example, a vertical presented between the third and fourth web result could be judged less relevant if it contains information that is redundant with web results 1-3. Second, the quality of the overall SERP may be diminished if the results from a highly salient vertical on the SERP (e.g., image vertical results) are very poor.

The main limitation of the SASI approach is that it is entirely retrospective. We can learn from assessors and determine trends that are likely to generalize across SERPs. However, we cannot directly reuse assessors judgements to evaluate new SERPs. Next, we discuss test-collection evaluation, where the goal is to create a portable test collection with relevance judgements that can be used to evaluate completely new SERPs (subject to certain layout constraints).

#### 4.2.2 Test-Collection Evaluation

Test-collection evaluation follows the Cranfield evaluation paradigm [Cleverdon, 1960], which is an important evaluation paradigm in ad-hoc document retrieval. In the case of ad-hoc retrieval, a test collection consists of: (1) a set of evaluation queries, (2) a corpus of documents, and (3) a set of human-produced relevance judgements indicating the binary or graded relevance of each document for each evaluation query. Given a test collection, ad-hoc retrieval systems can be evaluated using metrics that measure the system’s ability to rank documents in descending order of their relevance to the query. Test collections are portable, reusable, and allow us to compare different systems in a highly controlled environment.

In the context of aggregated search, a test collection consists of: (1) a set of evaluation queries, (2) a set of *cached* results from the different



verticals available to the system, and (3) a set of human-produced relevance judgements indicating the (binary or graded) relevance of each set of cached vertical results for each evaluation query. To date, aggregated search test collections have been built under the assumption that the aggregated search system is *not* responsible for deciding *which* results to retrieve from each candidate vertical. For this reason, human relevance judgements are gathered for the top results returned by each candidate vertical in response to the evaluation query. Given these three components, an end-to-end aggregated search system can be evaluated using metrics that measure the system’s ability to include the relevant verticals on the SERP and display them in a way that is consistent with their relevance to the query. Next, we review previously proposed methods that fit the general “test collection mold”.

**Arguello et al. [2011b]**. This methodology makes the following modeling assumptions. First, each vertical  $v \in \mathcal{V}$  is associated with some number ( $t_v$ ) of results that must be displayed if  $v$  is included on the SERP. Second, if vertical  $v$  is displayed on the SERP, then the system must display the top- $t_v$  results returned by  $v$  in response to query  $q$ . Third, all  $t_v$  results from the same vertical must be displayed together in a vertical “block” (either stacked horizontally or vertically, depending on  $v$ ). Fourth, vertical results can only be displayed in fixed locations relative to the web results (e.g., above the first web result, between the third and fourth web result, between the six and seventh web result, and below the last web result).

Given these presentation constraints, the aggregated search task can be cast as a *block ranking* task. Let  $\mathcal{B}_q$  denote the set of vertical and web blocks associate with evaluation query  $q$ .  $\mathcal{B}_q$  includes one vertical block for each vertical  $v \in \mathcal{V}$  that retrieves at least  $t_v$  results in response to query  $q$ , and one web block per sequence of web results that can not be split (e.g., web results 1-3, 4-6, and 7-10). Given query  $q$ , the goal of the aggregated search system is to produce a ranking of  $\mathcal{B}_q$  (denoted as  $\sigma_q$ ). The quality of  $\sigma_q$  is measured based on its similarity to an *ideal* or *reference* ranking of  $\mathcal{B}_q$  (denoted as  $\sigma_q^*$ ).

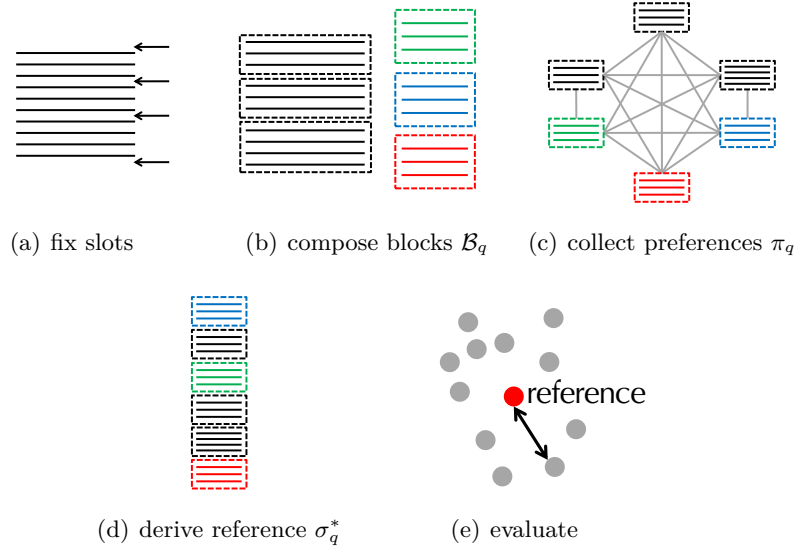
Two open questions remain: (1) How do we generate the reference block ranking  $\sigma_q^*$ ? and (2) How do we measure the similarity

between a predicted block ranking  $\sigma_q$  and the reference block ranking  $\sigma_q^*$ ? To address the first question, Arguello et al. [2011b] used crowdsourced workers to gather pairwise preference judgements on all pairs of blocks  $(b_i, b_j) \in \mathcal{B}_q$ . Then, the authors used the Schultz Voting Method [Schulze, 2011] to generate  $\sigma_q^*$  from these pairwise preferences. The Schultz Voting Method is designed to combine a set of pairwise preference judgements in order to score and rank elements in a set. To address the second question, Arguello et al. [2011b] used a variant of Kendall’s  $\tau$  [Kumar and Vassilvitskii, 2010] to measure the number of discordant pairs between blocks in  $\sigma_q$  and  $\sigma_q^*$ . Lower values of this metric indicate a greater similarity between  $\sigma_q$  and  $\sigma_q^*$  and are therefore better.

This methodology is explained graphically in Figure 4.3. The method assumes that vertical results can only be displayed in certain slots within the top web results (Figure 4.3(a)). Given query  $q$ , we first construct the web and vertical blocks associated with the query. In this case, we have three web blocks (due to having four slots) and three vertical blocks (Figure 4.3(b)). Then, we gather redundant preference judgements on all blocks pairs (Figure 4.3(c)). Next, we generate the ideal or *reference* presentation based on our pairwise block preferences (Figure 4.3(d)). Finally, we can evaluate any given presentation for this particular query by measuring its distance between the predicted presentation  $\sigma_q$  and the reference  $\sigma_q^*$  (Figure 4.3(e)).

Arguello et al. [2011b] used the Schultz Voting Method [Schulze, 2011] to convert web and vertical block preference judgements into a reference presentation, and used a variant of Kendall’s  $\tau$  [Kumar and Vassilvitskii, 2010] to measure the distance to the reference presentation. One could imagine other ways of using block-level relevance judgements to generate an idea block ranking and other ways of measuring distance or similarity with the reference.

**Zhou et al. [2012c].** The test collection methodology proposed by Zhou et al. [2012c] also treats aggregated search as a *block ranking* task. However, there are two main differences. First, the methodology gathers human relevance judgements on *individual* blocks of web or vertical results, rather than judgements on pairs of blocks. Second, the



**Figure 4.3:** Approach Overview.

methodology proposes an evaluation metric for *directly* measuring the quality of a particular block ranking  $\sigma_q$ , rather than evaluating based on the similarity between  $\sigma_q$  and an ideal ranking  $\sigma_q^*$ .

The proposed metric (referred to as *utility*) considers three important attributes of a block  $b_i$  within the context of a predicted block ranking  $\sigma_q$ : (1) the block’s relevance or *gain* with respect to the task ( $G(b_i)$ ), (2) the block’s examination probability ( $E(b_i)$ ), and (3) the cognitive effort required to process the results within the block ( $F(b_i)$ ). The *utility* of block ranking  $\sigma_q$  is given by:

$$\mathcal{U}(\sigma_q) = \frac{\sum_{b_i \in \sigma_q} (E(b_i) \times G(b_i))}{\sum_{b_i \in \sigma_q} (E(b_i) \times F(b_i))} \quad (4.1)$$

Equation 4.1 has the following intuition. The numerator can be viewed as the expected cumulative gain associated with block ranking  $\sigma_q$ . The gain of each web or vertical block  $b_i \in \sigma_q$  is discounted by the probability that a user will actually notice and examine the block. On the other hand, the denominator can be viewed as the expected effort in processing block ranking  $\sigma_q$ . The effort associated with each block

$b_i \in \sigma_q$  is also discounted by the block’s examination probability. The utility value is greater when  $\sigma_q$  provides the greatest gain (numerator) at the lowest cost (denominator).

In its raw form, utility is not necessarily in the range  $[0,1]$ , which makes averaging across queries tricky. Zhou et al. [2012c] suggested normalizing the raw utility value by dividing Equation 4.1 by the greatest utility attainable for query  $q$  and blocks  $\mathcal{B}_q$ .

Zhou et al. [2012c] proposed several alternatives for estimating  $E(b_i)$ ,  $G(b_i)$ , and  $F(b_i)$ .  $E(b_i)$  can be a combined function of the position of block  $b_i$  in  $\sigma_q$  and the visual salience of the block. For example, a block of image vertical results may have a higher examination probability than a block of web results (independent of its position).  $G(b_i)$  can be a combined function of the topical relevance of results within  $b_i$  and the query’s affinity for the vertical or source that produced  $b_i$ . For example, the query “pizza pics” has a high affinity towards the image vertical, while the query “pizza recipes” has a high affinity towards the web results. Finally,  $F(b_i)$  can depend on whether the surrogate representation includes image thumbnails, text, or a combination of both. Prior research considered the amount of effort required to make relevance assessments on surrogates containing different elements. Results suggest, for example, that surrogates augmented with images pulled from the underlying page can help users make more accurate and faster relevance judgements [Capra et al., 2013; Xue et al., 2008].

In follow-up work, Zhou et al. [2013b] proposed a variant of Equation 4.1 in which the examination probability of a particular block,  $E(b_i)$ , also considers the gains associated with blocks ranked above  $b_i$  in  $\sigma_q$ . This block examination probability estimate is motivated by the *rank-biased precision* (RBP) metric used in ad-hoc document retrieval [Moffat and Zobel, 2008]. The basic intuition behind RBP is that the likelihood that users will examine a particular result is not only a function of its rank, but also a function of the relevance associated with higher-ranked results. In other words, users are more likely to continue examining a ranked list if they are finding relevant documents.

**TREC 2013 and 2014 Federated Search Track.** The primary goals of the Federated Search Track was to evaluate algorithms for ver-

tical selection and results merging [Demeester et al., 2013, 2014]. The 2013 test collection includes 50 evaluation queries and the top-10 cached results from 157 different search engines, which include vertical-style search engines focused on different domains (e.g. games, health, recipes, sports) and different types of media (e.g., audio, images, videos), as well as a general Web search engine. All search engines correspond to online search services, and the top-10 results for each evaluation query were obtained by either screen-scraping or using APIs provided by the service. For each evaluation query, all top-10 search results from each search engine were assessed for relevance. Assessments were made using four different relevance grades: not relevant (0), relevant (1), highly relevant (2) and key (4). Finally, the test collection also includes sampled documents from each search engine. The Track organizers used 2000 single-term “queries” for sampling, which originated from different frequency-based bins from the vocabulary of the ClueWeb09-A test collection.<sup>1</sup> Track participants were encouraged to use these sampled documents to inform the vertical selection task.

For the 2013 Track, results merging (which corresponds to producing the end-to-end output) was evaluated as follows. Participants were allowed to merge the top-10 search results from all 157 search engines in an unconstrained fashion and the primary evaluation metric was NDCG@20. In other words, the ultimate goal for the system was to simply combine the top-10 search results from the different sources in a single ranked list, and to order the results in descending order of their graded relevance to the query. Results from the same non-web search engine were not required to be displayed together in a vertical block.

The Federated Search Track ran again in TREC 2014 and used a new test collection. The 2014 test collection includes 50 evaluation queries, the top-10 cached results from 149 different search engines, and 4000 sampled documents from each the available resources (twice as many as the 2013 test collection).

The 2014 results merging task was different from the 2013 results merging task in two respects. First, while results could still be merged in an unconstrained fashion, they could only originate from 20 differ-

---

<sup>1</sup><http://lemurproject.org/clueweb09/>

ent search engines. Thus, systems were required to predict at most 20 search engines to include in the merged ranking. Second, while the primary evaluation metric was NDCG@20, systems were also evaluated based on *intent-aware* NDCG@20 [Zhou et al., 2013b] (or IA-NDCG@20).

Intent-aware NDCG was originally proposed by Agrawal et al. [2009] for evaluating systems that diversify the search results from a single source. In the context of aggregated search, the idea behind IA-NDCG is to combine *document* relevance with *vertical* relevance. Let  $p_q^v$  denote the probability the vertical  $v$  is relevant to  $q$ . IA-NDCG@ $k$  is given by:

$$\text{IA-NDCG}@k = \sum_{v \in \mathcal{V}} p_q^v \times \text{NDCG}_v@k,$$

where  $\text{NDCG}_v@k$  is computed by only considering the top- $k$  documents originating from vertical  $v$ . Essentially, IA-NDCG@ $k$  considers the *weighted* average of NDCG@ $k$  values across all verticals. The weights are proportional to each vertical’s relevance to the query.

While the results merging task in the TREC 2013 and 2014 Federated Search Tracks involved unconstrained interleaving of results from different sources, both test collections can be used to evaluate systems that present vertical results in a *blocked* fashion [Bota et al., 2014]).

### 4.2.3 On-line Evaluation

On-line evaluation methods measure end-to-end system performance in a live environment using implicit feedback from real users. One type of on-line evaluation involves having a certain percentage of users use a baseline or (*control*) system and a different percentage of users use an *experimental* system for some period of time. The evaluation typically focuses on user interaction measures that are thought to be correlated with the quality of the user’s experience. For example, the evaluation might consider the percentage of queries without any clicks, which can be viewed as evidence of an unsuccessful search.

On-line evaluation has several advantages. First, it focuses the evaluation on real users in real situations. In this respect, on-line evaluation is ideal for testing systems that learn from individuals’ preferences and

behaviors and serve personalized results. For the same reason, it is ideal for testing systems that customize results based on the user’s context, for example, based on the current time and location. Second, on-line evaluation typically involves *lots* of users. In this respect, it ensures that the observed trends are likely to generalize across different populations.

On-line evaluation also has some limitations. First, implicit feedback is noisy. Two commonly used feedback signals in aggregated search are vertical *clicks* and *skips*. A vertical skip indicates that the user did *not* click the vertical, but clicked on a lower-ranked result. Clicks and skips are noisy indicators of relevance and non-relevance.<sup>2</sup> Users tend to click on results that are ranked higher on the SERP and results that are more visually salient. Moreover, users click results because they *perceive* the result to be relevant based on its surrogate representation, which may be misleading. Vertical skips are also noisy. In some cases, a user might extract valuable information from the vertical surrogate representation without clicking on it. The second limitation of on-line evaluation is that experiments are not repeatable. That is not to say that a hypothesis cannot be tested more than once. However, the users and the queries will be different, which means that the outcome measures will also be different. This can make it difficult to determine whether a particular approach has been re-implemented correctly.

**A/B Testing.** A/B testing evaluation involves having the experimental system respond to a small percentage of query traffic and evaluating the system based on measures derived from user interactions with the SERP [Jie et al., 2013; Ponnuswami et al., 2011b]. Prior work evaluated an end-to-end system by measuring the *click-through rate* (CTR) for each vertical independently.

The click-through rate can be measured two different ways. Jie et al. [2013] and Ponnuswami et al. [2011b] computed the click-through rate as the percentage of queries for which the vertical was displayed on the SERP *and* was clicked by the user. In contrast, Ponnuswami et al. [2011a] computed the click-through rate as the number of vertical clicks

---

<sup>2</sup>Interpreting implicit feedback is an on-going challenge in on-line evaluation and experimentation. For example, research shows that searchers who are struggling and searchers who are engaged in an exploratory task can exhibit similar search behaviors [Hassan et al., 2014].

over the total number of clicks *and* skips (Equation 2.6). In this case, a vertical skip requires a click on a lower-ranked result. This second version of the click-through rate is more conservative because we can be more confident that the user *viewed* the vertical and decided to not click it.

Another related metric that has been used in on-line evaluation is the *long dwell-time* vertical click-through rate [Jie et al., 2013]. This metric measures the percentage of vertical clicks where: (1) there were at least 100 seconds between the click and the next event on the SERP or (2) the click was the last event in the session. The idea behind this metric is to distinguish between productive clicks where the user found useful information and unproductive clicks where the user almost immediately returned to the SERP.

Evaluating based on the vertical clicks has one main drawback—the evaluation focuses on precision, but ignores recall. Obviously, we can not observe a click on a vertical that the system did not display. Therefore, it can be difficult to reliably estimate the number of false negative vertical predictions. For this reason, it is also useful to report each vertical’s *coverage*, which measures the percentage of queries for which the vertical was displayed (whether or not it was clicked). Coverage is related to, but not equal to recall. A system is said to improve over a baseline system if a vertical’s click-through rate increases and its coverage also increases or remains the same. Ponnuswami et al. [2011b] reported click-through rate and coverage for different verticals and different positions on the SERP.

**Random Output.** One limitation of A/B testing is that a new evaluation needs to be conducted each time we want to test a new experimental system. Conducting a new evaluation requires additional resources and risks degrading the experience of real users if the new system is unsuccessful. To address this limitation, Ponnuswami et al. [2011a] proposed a methodology that gathers implicit feedback on *randomly* generated aggregated SERPs (subject to certain layout constraints) and then performs the evaluation of an experimental system *retrospectively*.

The basic idea is the following. Suppose that verticals can only be



displayed in certain slots on the SERP (e.g., *top*, *middle*, *bottom*). During the data collection period, a certain percentage of query-traffic is shown randomly generated aggregated SERPs. In this case, the system displays all of the selected verticals in random slots with equal probability.<sup>3</sup> The system caches all the search results and user interactions for feature generation, model training, and evaluation. Let  $\mathcal{Q}$  denote the set of queries observed by the system during the data collection period. After the data collection period, a new experimental system can be evaluated retrospectively as follows. Suppose that we want to evaluate the system based on the click-through rate of the image vertical when it is slotted in the *top* position. Then, the experimental system can be evaluated by computing the click-through rate only on the subset of queries in  $\mathcal{Q}$  for which the experimental system *would have* displayed the image vertical in the *top* position.

Jie et al. [2013] and Wang et al. [2016] used the same method to evaluate end-to-end systems with fewer constraints. In this case, vertical results could be slotted above, below, and between any two web results. For a subset of queries  $\mathcal{Q}$  observed in a production environment, the system displayed randomly generated SERPs (subject to business constraints).<sup>4</sup> Finally, systems were evaluated *retrospectively* by considering only the subset of queries in  $\mathcal{Q}$  for which the system *would have* produced exactly the same randomly generated SERP.

Jie et al. [2013] and Wang et al. [2016] evaluated end-to-end systems using the following metric. Let  $k$  denote the set of items displayed on a particular SERP. Each clicked item receives a reward of  $-1$  and each skipped item receives a reward of  $+1$ . All items below the lowest-ranked click receive a reward of  $0$ . The total reward for a particular SERP is equal to the sum of rewards over all  $k$  items. Finally, systems were evaluated based on the average reward across evaluation SERPs.

**Interleaving.** Interleaving methods were originally developed for evaluating document ranking algorithms. The basic idea is to merge

---

<sup>3</sup>The approach assumes an upstream vertical selection component that selects those verticals that should be presented on the SERP.

<sup>4</sup>Commercial systems typically have business constraints that require the aggregated search system to present certain verticals at certain locations for a fraction of queries for which vertical returns results.

the results from two or more competing systems (omitting duplicates) and to measure performance using clicks. Each click on a document represents a “point” in favor of the ranking algorithm from which the document was selected. In the end, the best system is the one with the greatest number of clicks averaged across queries. A wide range of interleaving methods have been proposed in prior work [Chapelle et al., 2012; Hofmann et al., 2011, 2013; Radlinski et al., 2008]. Interleaving approaches are judged based on their ability to perform an *unbiased* evaluation and their ability to evaluate systems using as few queries as possible. To test whether an interleaving approach is unbiased, all competing rankers should end-up in a *tie* for a *randomly* clicking user.

Chuklin et al. [2013a] proposed an interleaving approach that can be used to interleave the results from two competing aggregated search systems. The proposed approach (Vertical-Aware Team Draft Interleaving, or VA-TDI) is an extension of the Team Draft Interleaving (TDI) approach used to evaluate rankers that return results from the same collection.

The TDI approach takes two rankings as input ( $\mathcal{A}$  and  $\mathcal{B}$ ) and returns a single interleaved ranking ( $\mathcal{I}$ ). The algorithm resembles the process of two sports team captains selecting players for their respective teams ( $\mathcal{T}_A$  and  $\mathcal{T}_B$ ). The two input rankings ( $\mathcal{A}$  and  $\mathcal{B}$ ) represent both captains’ priority lists of players. The two captains iteratively flip a coin to decide which captain chooses the next player. To avoid accidentally favoring one captain over the other, no captain can choose more than two players consecutively. When either captain chooses a player, they choose their *top* player that has not already been drafted.

The VA-TDI interleaving approach is similar to the original TDI approach [Chuklin et al., 2013a]. The algorithm proceeds normally until the first vertical result is appended to the interleaved list ( $\mathcal{I}$ ). Then, the algorithm only interleaves results from that same vertical  $v$  until the vertical block is formed—until there are no more results from vertical  $v$  in either  $\mathcal{A}$  or  $\mathcal{B}$ , or until the pre-determined block size ( $t_v$ ) has been reached. Chuklin et al. [2013a] present a simulation experiment that shows that VA-TDI was able to correctly predict the best aggregated search system with a reasonable degree of success, and also remained

impartial—the approach predicts a tie between rankers if exposed to *randomly* generated clicks.

### 4.3 Summary

In this chapter, we reviewed different evaluation methodologies for vertical selection and presentation.

Vertical selection evaluation is relatively straightforward. The goal of the system is to predict which candidate verticals are relevant to the query. In this respect, vertical selection is a multiclass classification task, and systems can be evaluated using metrics such as precision and recall (macro-averaged across queries or verticals).

The main challenge in vertical selection evaluation is obtaining the relevance label for a particular query-vertical pair. Prior work has considered using trained assessors with expert knowledge of the candidate verticals [Arguello et al., 2009b] or crowdsourced assessors whose redundant judgements can then be combined into gold standard judgements using a majority vote [Zhou et al., 2012c].

Relevance labels can also be derived from vertical clicks and skips in an on-line or live environment. In this case, a vertical click suggests a true positive prediction and a vertical skip (a click on a lower-ranked result, but not the vertical) suggests a false positive prediction. Deriving relevance labels from vertical clicks and skips has one main challenge—the system cannot observe clicks if the vertical was not selected. In this respect, it is easy to measure precision, but difficult to measure recall. One solution is to evaluate the vertical selection system *retrospectively*. In this case, the system always displays the vertical in the same position for some fraction of query traffic. Then, a vertical selection system can be evaluated based on its ability to correctly predict users’s clicks and skips.

End-to-end evaluation, which involves vertical selection and presentation, is complicated by the fact that the system needs to make more decisions. The system must decide which candidate verticals to display, where, and possibly even how. Whole-page evaluation methods gather quality judgements retrospectively. Assessors are shown SERPs pro-

duced by the system and rate the quality of the output along different dimensions. Test collection evaluation methods use a set of evaluation queries, relevance labels on the results produced by the different candidate verticals, and evaluate using metrics that are thought to be correlated with user satisfaction. Finally, on-line methods evaluate using clicks from real users in a live environment.

In terms of on-line methods, a very principled approach is interleaving. The general idea is to merge the output from two or more competing aggregated search systems and to evaluate using clicks. The system with the greater number of clicks is deemed the most effective.

# 5

---

## Search Behavior with Aggregated Search

---

User studies are essential for understanding how real users interact with information retrieval systems, including aggregated search systems. In a general sense, running a user study involves exposing participants to different experimental conditions and measuring differences in one or more outcome measures. User studies aim to understand what factors influence user behaviors and how. Because user studies are conducted in a controlled setting, the researcher can manipulate different characteristics of the system, the search task, or the search context. Moreover, the researcher can target users with different characteristics, for example, users with different levels of search experience or different cognitive abilities. Likewise, the researcher can study the effects of a particular manipulation on both *objective* measures of performance (e.g., the time to task completion), as well as *subjective* measures of performance (e.g., participants' perceived level of system support).

In the context of aggregated search, user studies have been conducted to answer two main questions: (1) What do you users want from an aggregated search system? and (2) What are different factors that affect users' behaviors and experiences? With respect to the first question, prior studies have focused on validating different metrics used

in aggregated search evaluation. With respect to the second question, prior studies have considered how different characteristics of the system, the user, the search task, or the search context influence search behaviors and effectiveness.

## 5.1 Evaluation Metric Validation

In Section 4.2.2, we discussed two test collection evaluation methods proposed in prior work [Arguello et al., 2011b; Zhou et al., 2012c]. As a reminder, Arguello et al. [2011b] proposed an evaluation methodology that considers the similarity between a *predicted* aggregated SERP and an ideal (or *reference*) aggregated SERP for the given query. Zhou et al. [2012c] proposed a utility-based evaluation metric that considers three aspects of verticals displayed on the SERP: (1) the vertical’s relevance to the query, the vertical’s examination probability (based on its position and visual representation), and the cognitive effort required to process the vertical results. Both of these evaluation methodologies were validated with user studies.

One might argue that a good evaluation metric is one with a high level of agreement with users’ preferences. In other words, aggregated SERPs that are preferred by users should be scored as being superior by the metric. In both studies, the researchers measured the level of agreement between the proposed metric and participants’ preference judgements on pairs of aggregated SERPs displayed side-by-side. Both studies found three important trends.

First, the level of agreement between study participants was far from perfect. Agreement was measured in terms of Fleiss’ Kappa ( $\kappa_f$ ), which corrects for the expected agreement due to random chance [Fleiss, 1971]. Agreement between participants was about 20%. This level of agreement is better than random agreement (i.e.,  $\kappa_f = 0\%$ ), but is still fairly low. Thus, one would never expect an evaluation metric to predict preference behavior for individual users 100% perfectly.

The second important trend is that both metrics agreed with a majority vote preference at a level of about 60%, which is well above

random. In both experiments, pairs of aggregated SERPs were rated by multiple assessors, and these assessments were combined into final preference using a majority vote. If we consider a majority vote preference as representative of the “average user”, then both metrics predicted the preference of this “average user” with some degree of success.

The final important trend is that agreement between the metric and the majority vote preference was higher for pairs of aggregated SERPs with very different metric values. In other words, agreement was high for pairs where the metric value for one SERP was high and the other was low. Agreement was lower for pairs of aggregated SERPs with similar metric values. Put differently, both metrics were relatively poor in predicting preferences between two SERPs of similar quality (i.e., middle vs. low and middle vs. high).

From both of these studies, we can determine that users’ preferences on pairs of aggregated SERPs are not random and can be modeled using an evaluation metric to some extent. That said, there is room for improvement. Developing evaluation metrics that achieve a higher level of agreement with users’ preferences is a promising direction for future work.

## **5.2 Studies Supporting Vertical Selection and Presentation**

Current aggregated search systems are defined by three important design considerations.

First, current systems predict which verticals are relevant to a query and display a few of the top results from each selected vertical alongside the core web results. The goal of the system is to “showcase” verticals that may be useful to the current user. An alternative to combining results from different sources in a single presentation would be to simply provide access to the web and vertical results using tabs. Users could then click on different tabs to see results from a specific source.

Second, current systems display the results from the same vertical together in a vertical block (either stacked horizontally or vertically). An alternative would be to interleave results from different sources in a completely unconstrained fashion as is typically done in text-based

federated search.

Finally, current systems construct the aggregated SERP dynamically. In other words, current systems attempt to display *only* the relevant verticals and display the most confidently relevant verticals in a more salient manner, for example, by presenting them higher on the SERP. An alternative would be to show results from all verticals (or only the most relevant) in *fixed* positions.

Next, we discuss user studies that lend support to the current paradigm.

**Aggregated vs. Tabbed.** One alternative to the current paradigm would be to have users switch between results from different sources using tabs. However, studies have found that integrating results from different sources in a single presentation has at least three main benefits: (1) it reminds users that a particular vertical has relevant content, (2) it provides easy access for users who want results from different sources, and (3) it raises awareness about the contents in a particular vertical, which can help users in future searches.

Several studies have found that users are more likely to click on and bookmark vertical content when it is blended into the core web results [Arguello et al., 2012; Sushmita et al., 2009; Turpin et al., 2016]. This result suggests that users do not always know *a priori* that a particular vertical has relevant content and may therefore benefit from seeing vertical results without explicitly requesting them.

Sushmita et al. [2010a] present a query-log analysis that shows that users often click on results from different sources for non-navigational searches (searches with more than one click). Bota et al. [2015] report on a user study that asked participants to construct “bundles” of relevant search results associated with different aspects of a multi-faceted search topic. Specifically, participants were asked to imagine that they were preparing a blog post on a particular topic and were given access to an aggregated search system in order to find relevant content. Results found that about 80% of all bundles had results from different sources, suggesting that users may benefit from seeing results from different sources on the same SERP.

Bron et al. [2013] investigated search behavior across multiple ses-



sions with a system that allowed users to switch between an aggregated view, which combined results from different sources, and a source-specific view, which allowed users to restrict the search results to a single source. Results found that the aggregated view helped raise awareness about the contents in each source, and influenced participants to explore different sources in later search sessions. Thus, aggregated interfaces may have long-term benefits in helping users understand the different resources available to them.

Together, the results from these studies suggest users may benefit from seeing results from different sources on the same presentation.

**Constrained vs. Unconstrained Interleaving.** Current systems display the results originating from the same vertical together in the form of a vertical block (stacked horizontally or vertically). An alternative strategy would be to interleave the results from different sources in an unconstrained fashion. Results from different sources could simply be merged into a single ranked list. To date, no single user study has directly compared these two different ways of presenting aggregated search results. However, prior research suggests that users would prefer a grouped display.

Early work in psychology developed the Gestalt principles of pattern recognition [Koffka, 1935]. These principles explain how people perceive groups of objects in an information display, such as an aggregated SERP. For example, the Gestalt principles of *similarity* and *proximity* state that items in a display that are visually similar or closer together are perceived as a group. More recently, Palmer [1992] proposed the principle of *common region*, which states that items displayed in a common region (such as within a border or a different-colored background) are perceived as a group.

Based on the Gestalt principle of *proximity*, it seems logical that grouping same-vertical results together communicates to the user that they are the same *type* of result and originate from the same source. In this respect, a grouped display may help users more quickly identify results from a relevant vertical and ignore results from a non-relevant vertical.

While this has not been empirically studied in the context of aggre-

gated search, studies suggest it is a sensible hypothesis. Nygren [1996] report on an early study of users scanning a webpage in search for a specific piece of information. Results found that participants were quicker at finding items on the page in cases where similar items were grouped together. More closely related to information retrieval, Dumais et al. [2001] evaluated six different search interfaces that were augmented with category information. The goal was to help users identify relevant documents by displaying the topical category of each search result along with its title and summary snippet. Half of the interfaces were *grouped* (results from the same category were displayed together) and the other half were *ungrouped* (results from the same category were not displayed together). Participants used both interfaces to search for relevant documents for a given task. Results found that participants completed search tasks quicker when using all three grouped interfaces than when using all three ungrouped interfaces.

**Static vs. Dynamic.** The current paradigm is to construct the aggregated SERP dynamically for each query. Specifically, current systems aim to only display those verticals that are relevant and to display the most relevant verticals higher on the SERP. Another alternative to this current paradigm would be to combine results from all the different sources in a static, query-independent layout.

Several studies have found that users prefer to not see results from non-relevant verticals on the aggregated SERP. Arguello et al. [2012] and Turpin et al. [2016] experimented with an aggregated interface that always presented results from four different verticals in fixed positions on the SERP *irrespective of the query*. Study participants rated the system poorly compared to a system that only provided access to the different sources using tabs.

Sushmita et al. [2010b] present a study where participants were given search tasks and access to an aggregated SERP. The study manipulated three variables: (1) the vertical presented (image, news, video), (2) the vertical's position on the SERP (top, middle, bottom), and (3) the vertical's relevance to the search task (high, medium, low). Results found that participants preferred SERPs in which the more relevant verticals were positioned higher on the SERP.

More recently, Chen et al. [2015] measured user satisfaction with relevant versus non-relevant vertical results presented on the SERP. Participants in the study were asked to complete search tasks using pre-constructed queries and SERPs, and reported their satisfaction with SERPs using a 5-point scale. The authors experimented with five different types of verticals and three different positions on the SERP. Non-relevant vertical results were produced by altering the query sent to the vertical search engine.

The results from this study found several interesting trends. First, as one might expect, participants reported greater levels of satisfaction with SERPs that presented relevant versus non-relevant vertical results. Second, the effect was stronger for salient verticals such as images. Third, the effect was stronger when the vertical was displayed higher on the SERP.

### **5.3 Factors Affecting Vertical Results Use and Gain**

Several studies have investigated different factors affecting user engagement with results from different sources aggregated on the SERP. Specifically, prior work has focused on how different factors of the vertical, the search task, and the user can affect user engagement with vertical results aggregated on the SERP. In terms of the vertical, studies have found that the vertical's relevance, position, and presentation affect user engagement. In terms of the search task and the user, studies have found that search task's complexity and the user's perceptual speed (a type of cognitive ability) can also have an effect.

**Vertical Relevance.** As might be expected, studies have found that relevance increases user engagement with the vertical results. As mentioned above, Arguello et al. [2012] and Turpin et al. [2016] experimented with interfaces that always displayed results from the same verticals (in fixed positions) irrespective of the query. Because the verticals were always displayed, in many cases they were not relevant to the search task. Study participants reported a preference towards interfaces that did not showcase vertical results on the main SERP.

Chen et al. [2015] measured user satisfaction with aggregated

SERPs with relevant vertical results versus non-relevant vertical results, generated by altering the query before issuing it to the vertical. Participants reported greater levels of satisfaction with SERPs that presented relevant vertical results. Liu et al. [2015] report on an eye-tracking study that measured percentage of eye fixations on different results on SERPs with relevant versus non-relevant vertical results. Again, non-relevant vertical results were generated by altering the query issued to the vertical. More visual attention was given to verticals that presented relevant results. However, this effect was weaker for visually salient verticals. For highly salient verticals, the amount of visual attention was less influenced by the relevance of the vertical results.

**Vertical Position.** Studies have also found that the vertical's position influences user engagement with the vertical results. In general, users are more likely to engage with vertical results that are presented higher on the SERP. Chen et al. [2015] experimented with relevant versus non-relevant results displayed in three positions on the SERP. Participants reported higher levels of satisfaction for SERPs that displayed relevant vertical results higher on the SERP (i.e., rank 1 vs. rank 3 vs. rank 5), and *lower* levels of satisfaction for SERPs that displayed non-relevant vertical results lower (i.e., rank 5 vs. rank 3 vs. rank 1).

**Vertical Presentation.** Prior research has also found that user engagement (as measured by clicks and visual attention) is also biased by the way results are presented on the SERP. Even with SERPs that present results of the same type (e.g., web results displayed using the page title, summary snippet, and URL), users are influenced by different presentation decisions. For example, Yue et al. [2010] found a click-through bias in favor of textual snippets with bolded versus non-bolded query-terms. Hofmann et al. [2012] focused on the task of predicting clicks on web search results. The authors generated different features from the textual surrogate representation, including the title, snippet, and url length, as well as the presence of bolded query-terms in each component. These attributes of the surrogate representation were found to be predictive of users' clicks.

Within the context of aggregated search, studies have also found

that users click on vertical results that are more visually salient. Sushmita et al. [2010c] found a click bias in favor of image results and Sushmita et al. [2010b] found a click bias in favor video results. Finally, as mentioned above, Liu et al. [2015] found greater levels of visual attention given to more salient verticals (e.g., images) irrespective of relevance.

Together, these results show a complex interplay between vertical relevance, position, and presentation. All three influence user engagement with the vertical results, either measured in terms of clicks or eye fixations. However, there are interaction effects. For visually salient verticals, relevance and position seem to matter less. This has important implications for aggregated search evaluation and prediction. A vertical's position on the SERP may matter less for verticals that are more visually salient. Similarly, evaluation methods for vertical selection may need to more severely punish false positive mistakes for highly salient verticals.

**Task Complexity.** A search task's *cognitive complexity* refers to the amount of learning and cognitive effort required to complete it. A simple search task might require finding or verifying a particular fact, for example: What is the name of the deepest part of the ocean? In contrast, a complex task might require comparing/contrasting a set of items along a number of different dimensions and making a priority list or recommendation, for example: Which hybrid SUV would you to buy based on criteria such as price, gas mileage, and warranty?

Prior work has found that more complex tasks are associated greater levels of search interaction [Aula et al., 2010; Liu et al., 2010a,b, 2012; Wu et al., 2012]. For example, more complex tasks are associated with a greater number of queries, clicks, and bookmarks; a greater number of queries without a click and clicks without a bookmark; longer landing-page dwell times; and longer completion times. Additionally, Jansen et al. [2009] observed that participants completing more complex tasks engaged with search results from a wider range of sources.

In the context of aggregated search, Arguello et al. [2012] investigated whether participants completing more complex tasks engaged more with content from different verticals. Participants were given tasks

of varying levels of complexity and interacted with two different systems: a *blended* system that combined results from different verticals in an aggregated SERP and a *tabbed* system that only provided access to the different verticals using tabs. While the effect was subtle, participants interacted more with results from different verticals while completing more complex tasks, but only with the blended system.

**Perceptual Speed.** A person's *perceptual speed* (PS) refers to their "speed in comparing figures and symbols, scanning to find figures or symbols, or carrying out other simple tasks involving visual perception." [Ekstrom et al., 1979]. Several studies have found that low-PS searchers experience greater workload and interact at slower rates than high-PS searchers [Al-Maskari and Sanderson, 2011; Brennan et al., 2014].

In the context of aggregated search, Turpin et al. [2016] investigated the effects of perceptual speed on search behavior and performance. Low-PS and high-PS participants interacted with two different systems: one that blended vertical and web results into the main SERP (*blended*) and one that only provided access to different verticals using tabs (*tabbed*). High-PS participants rated both systems and their own performance as higher than low-PS participants. Furthermore, high-PS participants spent similar amounts of time completing tasks with both interfaces, but low-PS participants took significantly *longer* with the blended interface. This result sets into question a "one size fits all" aggregated search solution, and suggests that individuals with different cognitive abilities may benefit from different ways of displaying aggregated search results.

## 5.4 Spillover Effects in Aggregated Search

In the context of vertical selection, a false positive prediction happens when the system displays a particular vertical that is not relevant to the user. For example, suppose that a user enters the query "tesla" because they want to find biographical information about Nikola Tesla. Presenting image vertical results might be considered a false positive prediction because they do you satisfy this particular user's information

need.

Aggregated search evaluation methods typically assume that all false positive predictions are *equally* bad. However, let us consider two different situations. In the first situation, the system displays images of Nikola Tesla, and in the second situation, the system displays images of “tesla” the electric car. Would these two sets of non-relevant image results equally degrade the user’s experience? Seeing images about the unintended query-sense (i.e., the electric car) might cause the user to have less confidence in the *other* results presented on the SERP.

Several studies have investigated how the results from one particular source can affect user engagement with results from *other* sources aggregated on the SERP [Arguello and Capra, 2012; Arguello et al., 2013; Arguello and Capra, 2014; Arguello, 2015; Bota et al., 2016]. This phenomenon has been referred to as the “spillover” effect. Thus far, prior work has focused exclusively on the spillover effect in the context of ambiguous queries such as “tesla”, and most of this work has focused on understanding how the query-senses represented in a set of vertical results can influence user engagement with the core web results [Arguello and Capra, 2012; Arguello et al., 2013; Arguello and Capra, 2014; Arguello, 2015]. As an exception, Bota et al. [2016] investigated how the query-senses represented in an entity card displayed on the SERP can influence user engagement with the web results. An *entity card* or *knowledge graph display* provides background information about an entity associated with the query and may include images, links to related pages, and links to related queries.

Results from all these studies suggest that users are more likely to engage with the web results when the query-senses associated with the vertical results (or the entity card results) are more consistent with the user’s intended query-sense. For example, a user looking for information about “tesla” the scientist is more likely to engage with the web results if the image vertical results contain pictures of the scientist versus the car.

Prior work has also investigated how different factors influence the level of spillover from the vertical to the web results, including the vertical’s visual salience, the vertical’s position on the SERP, and whether

the vertical is displayed in a way that distinguishes it from the other results on the SERP. In all these studies, the level of spillover was measured by manipulating the vertical results and observing the differences in participants' interactions (e.g., clicks, bookmarks, mouseovers) with the web results on the SERP.

In all these studies, the spillover was measured by measuring user engagement with other results on the SERP using clicks, bookmarks, and other measures.

**Vertical Salience.** Several studies have found that the spillover effect is stronger for verticals that are more visually salient [Arguello and Capra, 2012; Arguello et al., 2013; Arguello and Capra, 2014; Arguello, 2015]. For example, the spillover effect is stronger for the images vertical than the news vertical. An eye-tracking study by Liu et al. [2015] found that images attracted participants' visual attention more than news results. Thus, image results may cause more spillover because participants are more likely to notice that query-senses in the image results and then assume that the other sources on the SERP are also skewed towards the same query-senses.

Arguello et al. [2013] investigated this same factor by considering two different versions of the shopping vertical: one that included a thumbnail image of each product displayed in the vertical block (in addition the title and price) and one that did not include a thumbnail image. The version that included a thumbnail image (i.e., the more salient version) had a slightly more spillover, although the effect was not significant.

**Vertical Position.** Prior results also suggest that the vertical's position on the SERP can affect the level of spillover. For example, the level of spillover is greater when the vertical is displayed above the web results versus somewhere in the middle [Arguello and Capra, 2014] or to the right side of the web results [Arguello and Capra, 2016]. Displaying the vertical above the web results may have caused a stronger spillover effect for two possible reasons: (1) perhaps participants were more likely to notice the vertical results, or (2) perhaps participants assumed that the system as a whole was more confident about the query-senses in the vertical results.



**Vertical Layout.** As mentioned above, the Gestalt principle of *common region* states that items displayed in a common region, such as within a border or with a different-colored background, tend to be perceived as a group [Palmer, 1992]. In the context of aggregated search, vertical results are typically displayed on the SERP without a strong visual cue to distinguish them from other results on the SERP. In contrast, for example, advertisements on the SERP are typically displayed using a different-colored background.

Prior work investigated whether displaying the vertical results enclosed in a border reduces the level of spillover [Arguello and Capra, 2014]. Including a border had a subtle moderating effect—for some verticals, the level of spillover was slightly greater without a border than with a border.

In a follow-up study, Arguello and Capra [2016] found an interesting additive effect between the vertical's position and the presence of a border—the highly salient image vertical results had almost no spillover when presented to the right side of the web results and enclosed in a border. This result suggests an interesting additive effect from the Gestalt principles of proximity and common region—the images had little spillover when presented away from the web results and with a border to distinguish them as a different group.

#### 5.4.1 Improving Aggregated Search Coherence

Incoherent aggregated search results are likely to occur when the query is ambiguous and the results from different sources aggregated on the SERP are skewed towards different senses. A natural question is: How often does this happen? With respect to query ambiguity, Sanderson [2008] conducted an analysis of a commercial search engine's query-log, and found that about 4% of all unique queries and 16% of all unique head queries had an exact match with an ambiguous entity in Wikipedia or an ambiguous concept in WordNet. With respect to different sources being skewed towards different senses, an analysis by Arguello and Capra [2016] found that the top results from four different verticals were typically skewed towards one particular sense of an ambiguous query. By comparison, the top web results appeared to be more

diversified across different query-senses. Santos et al. [2011] found the same trend in the top queries issued to different verticals. For example, the image vertical had mostly queries about “amazon” the rainforest, and the shopping vertical had mostly queries about “amazon” the company.

Arguello [2015] proposed a simple method for improving the level of coherence between the vertical and web results. Broadly speaking, the approach proceeds in two steps: (1) cluster the top web results based on their query-sense similarity and (2) iteratively select vertical results that are similar to different web clusters. Vertical and web results were represented using the prediction confident values from about 200 topical classifiers trained on ODP data.<sup>1</sup> Results found that improving the level of coherence between the vertical and web results influenced study participants to make more correct decisions about their level of engagement with the web results—to engage with the web results when at least one of them was relevant, and the *avoid* engaging with the web results when none of them were relevant.

## 5.5 Scanning Behavior in Aggregated Search

Understanding how users scan search results is critical for IR evaluation. One might argue that the goal for an IR system is to present the most confidently relevant material in areas where the user is most likely to examine it. In the context of a “ten blue links” interface, where users typically scan results top-to-bottom, this boils down to ranking documents by their probability of relevance.

In the context of aggregated search, the scanning behavior of users is less predictable for several reasons. Aggregated search systems combine results with very different surrogate representations (e.g., web results, images, videos, news, local business, etc.). Certain results are more visually salient than others. Furthermore, in many current aggregated search systems, results are not only stacked vertically from top to bottom. Instead, it is now common practice to display vertical results or entity cards (also referred to as knowledge graph displays)

---

<sup>1</sup><https://www.dmoz.org/>

to the right side of the web results (also referred to as the *two-column* format [Navalpakkam et al., 2013]). Knowing how users will scan a particular SERP can help us determine how to reward (or *penalize*) the presentation of relevant (or *non-relevant*) web or vertical results in a particular location on the SERP.

Next we review three areas of related work. Research on *click modeling* aims to estimate the probability that a user will click a result presented in a particular position on a SERP. Research on *mouse movement modeling* aims to predict mouse movement patterns. Prior work shows that, under certain conditions, eye gaze and mouse cursor movements are highly correlated [Huang et al., 2011, 2012a,b]. Thus, predicting mouse cursor movements on a SERP can help predict how a user might visually scan the page. Finally, *visual attention modeling* attempts to directly predict where the user is looking based on mouse cursor data and characteristics of the SERP. In all three areas, we focus mostly on research that directly addresses aggregated SERPs with heterogeneous results (possibly using a non-linear layout).

**Click Models.** Click models attempt to estimate the probability that a user will click a particular search result displayed on position  $i$  on a SERP. This probability is denoted as  $P(C_i = 1)$ . Click models have received considerable attention in prior work partly because clicks are *observed* by the search engine. In this respect, a click model can be easily evaluated based on how well it predicts clicks from real users. A commonly used metric for evaluating a click model is *perplexity*, which is inversely proportional to the probability assigned by the model to a series of *observed* clicks.

A click model can be used as part of an evaluation metric. For example, Chuklin et al. [2013b] proposed the following *utility-based* metric:

$$\text{UMB} = \sum_{i=1}^n P(C_i = 1) \times r_i,$$

where  $n$  denotes the number of items on the SERP and  $r_i$  denotes the relevance of the item in position  $i$ , which can be estimated using graded judgements as  $r_i = \frac{2^{G_i} - 1}{2^{G_{max}}}$  [Markov et al., 2014]. The idea behind this utility-based metric is to place more emphasis on the relevance of items

more likely to be clicked. The click model can consider different factors, including the item's predicted relevance, its position, the predicted relevance of nearby items, and the visual salience of nearby items.

With respect to click-behavior in the presence of vertical results on the SERP, prior research has found four important trends.

First, users tend to click more on a vertical result than on a web results displayed in the same position [Chen et al., 2012; Diaz et al., 2013; Wang et al., 2013]. One possible explanation is that current systems are mostly successful in displaying vertical results when they are relevant to a query. Another explanation is that vertical results are more visually salient and attract more clicks independent of relevance.

Second, users are more likely to click on web results that are in close proximity to highly salient vertical results such as images and videos [Chen et al., 2012].

Third, clicks on highly salient verticals tend to be the last click on the SERP [Chen et al., 2012]. In this case, the authors argue that highly salient verticals allow users to make more correct click decisions based on the surrogate representation. In this respect, vertical clicks tend to ultimately result in the user being satisfied.

Finally, results show that if the first of multiple SERP clicks is on a vertical result, users tend to subsequently click on a higher-ranked web result if one is available [Wang et al., 2013]. The authors argued that users tend to skip web results to click on a lower-ranked vertical result that is more visually salient. In cases where the vertical result does not satisfy the information need, users tend to return to scanning the web results top-to-bottom.

Several click models have been proposed that consider the presence of vertical results on the SERP. Chen et al. [2012] proposed a click model that considers the rank of the item, its visual salience (depending on the originating source), and its distance from the vertical results presented on the SERP. Wang et al. [2013] proposed a click model that also favors clicks on results that are in close proximity to a vertical, but favors clicks above the vertical more than below. Markov et al. [2014] proposed a click model that also favors clicks on results that are close to the vertical, but allows the model to estimate different proba-

bilities for results above or below the vertical. To date, these different vertical-aware click models have not been evaluated against each other. However, they all outperform click models that do not consider vertical results on the SERP.

**Mouse Movement Models.** In the context of a “ten blue links” interface, prior work shows a correlation between mouse cursor position and visual gaze. Interestingly, however, Huang et al. [2012a] show that the strength of the correlation depends on the user, the search task, and the time spent on the SERP. Moreover, Huang et al. [2011] also show that mouse movement patterns can help predict the relevance of a particular result and can help distinguish between good and bad abandonment [Huang et al., 2011].<sup>2</sup>

In the context of aggregated search, Diaz et al. [2013] proposed a model for predicting mouse cursor transitions between non-overlapping bounding boxes, or *modules*, around different SERP components (e.g., individual web results, vertical blocks, advertisements, query suggestions, and logos). Here, a *transition* required the mouse to remain stationary for a certain period of time in the two modules. In a motivating analysis, Diaz et al. [2013] show that aggregated SERP configurations from two commercial search follow Zipf’s law. In other words, few configurations are very frequent and most configurations are very rare. For this reason, the authors proposed a model for predicting mouse transitions that can generalize to previously *unseen* configurations.

The proposed approach uses machine learning to train a model to predict transitions between modules  $m_i$  and  $m_j$  as a function of a set of features. Features included characteristics of both modules, such as their size and identity (e.g., web, vertical identity, logo, etc.), as well as features generated from the module-pair, such as their distance and whether they contain the same result-type. Training data was generated from module transitions observed from real users—each transition from module  $m_i$  to  $m_j$  was considered a *positive* instance and each *absence* of a transition from module  $m_i$  to  $m_k$  was considered a *negative* instance.

A model trained to predict module transitions can be used in differ-

---

<sup>2</sup>Good abandonment happens when the user does not click on a result, but is satisfied by the information presented on the SERP.

ent ways. For example, given a new SERP, we can predict the transition probabilities between all pairs of modules on the SERP. Let  $\mathbf{P}$  denote the transition matrix predicted by the model. Then, we can predict the probability of *being in particular module* on the SERP (denoted as  $\pi(m_i)$ ) by taking powers of this transition matrix,  $\pi = \mathbf{P}^k \mathbf{e}$ , using a large value of  $k$ . Finally, assuming a correlation between mouse cursor and eye fixations,  $\pi(m_i)$  can be treated as the amount of visual attention given to module  $m_i$  within the context of the SERP.

**Visual Attention Models.** As previously mentioned, the extent to which mouse position correlates with eye gaze position depends on different factors of the user, the task, and the time spent on the SERP [Huang et al., 2012a]. Using mouse movement and eye tracking data collected from a controlled study, Navalpakkam et al. [2013] proposed different models for predicting eye gaze position from mouse activity. Specifically, the authors focused on two different tasks: predicting the  $y$ -coordinate of eye gaze (a regression task) and predicting the current module being examined (a multiclass classification task). For both tasks, the authors included features such the time spent on the SERP, the current mouse position, the current mouse velocity (magnitude and direction), the total mouse distance so far on the SERP, and the result-type of the module associated with the current mouse position. For both predictive tasks, the authors trained a *global* model (trained on the combined data from all study participants) and a *user-specific* model (trained on the each individual participant’s data). In both tasks, the user-specific model outperformed the global model.

Liu et al. [2016] developed and evaluated different models for predicting visual attention given to different elements on the SERP (e.g., individual web results, vertical result blocks, etc). Specifically, the authors focused on two different tasks: (1) predicting the percentage of total fixation durations associated with a particular element (a regression task), and (2) predicting whether a user will fixate on a particular element at least once (a binary classification task). Here, the unit of analysis was the individual page element. In other words, training and test instances corresponded to page elements within a particular SERP displayed in the study. The authors experimented with two dif-

ferent types of features: (1) content salience features (e.g., the element type, position, size, text-to-area ratio, font size), and (2) visual salience features inspired from research in the field of visual processing. Visual salience features considered visual properties of the SERP element such as its color, intensity, and orientation. Interestingly, the authors did not consider user interaction features at all. As one might expect, visual salience features were especially predictive when the SERP included vertical results with images. Furthermore, visual salience features were also useful for predicting the amount of visual attention given to individual elements within a vertical block (e.g., individual image results within an image vertical block).

Lagun and Agichtein [2015] proposed an algorithm for predicting eye fixation position on SERPs and other types of webpages. Broadly speaking, the algorithm proceeds in three steps. First, the page is segmented into regions of interest (e.g., titles, headings, paragraphs, images, navigational bars). Then, the model predicts the eye gaze position relative to each region of interest as a function of two types of features: (1) interaction features derived from current mouse and scroll position and movement, and (2) content salience features derived from the region of interest (e.g., size, element type, font size, text-to-area ratio). Finally, the different region-specific predictions are combined to predict the current eye fixation position on the SERP. Results found that combining interaction and content salience features improves prediction performance over using each feature type in isolation across different domains (e.g., aggregated SERPs, as well as news, shopping, and social network websites).

# 6

---

## Special Topics in Aggregated Search

---

### 6.1 Domain Adaptation for Vertical Selection

Training a vertical selection model requires training data, either in the form of human-produced vertical relevance judgements or in the form of vertical clicks and skips from real users in the operational setting. In the field of machine learning, *domain adaptation* is the task of using training data from one or more domains (referred to as the *source* domains) to learn a model that can make predictions on another domain (referred to as the *target* domain). In the context of aggregated search, prior research investigated the use of domain adaptation techniques for the purpose of vertical selection [Arguello et al., 2010]. Specifically, the goal was to use training data associated with a set of existing verticals (referred to as the *source* verticals) to learn a model that can make vertical selection predictions for a new vertical (referred to as the *target* vertical). In this case, the training data associated with each source vertical corresponded to a set of queries with human-produced vertical relevance judgements.

Arguello et al. [2010] focused on two model properties for the purpose of domain adaptation: *portability* and *adaptability*. A portable model is one that can make predictions for any target vertical, and an



adaptable model is one that can be trained to make predictions for a specific vertical.

**Model Portability.** As we have seen so far, approaches for vertical selection use machine learning to combine a wide range of features. Certain features (referred to as *portable* features) tend to have a consistently *positive* or *negative* relationship with vertical relevance *irrespective* of the exact vertical being considered. In contrast, other features (referred to as *non-portable* features) have an inconsistent relationship with vertical relevance across different verticals.

Let us consider the difference between portable and non-portable features with some examples. One type of feature might consider the similarity between the input query and those queries recently issued directly to the vertical by users. This is an example of a portable feature because it is likely to be *positively* correlated with relevance irrespective of the vertical being considered. That is, the higher its value, the more relevant the vertical from which the feature was generated. Another type of feature might consider the likelihood that the query is a navigational query, which is better addressed by displaying web versus vertical results. This is also an example of a portable feature because it is likely to be *negatively* correlated with relevance irrespective of the vertical in question. On the other hand, consider a feature that describes whether the query is related to the travel domain. This is an example of a non-portable feature because it is likely to be positively predictive for a travel-related vertical, but negatively predictive for a vertical focused on a different domain.

The trick to learning a portable model is to focus on portable features. Arguello et al. [2010] experimented with three different approaches for learning a portable model. One approach is train a model by combining all the training data from every source vertical. Let  $\mathcal{S}$  denote the set of source verticals and let  $\mathcal{Q}_v$  denote the set of queries with (positive and negative) relevance labels with respect to source vertical  $v \in \mathcal{S}$ . In this and the next two approaches, the joint training set is defined as:

$$\mathcal{Q}_* = \cup_{v \in \mathcal{S}} \mathcal{Q}_v.$$

In Chapter 2, we distinguished between query, vertical, and query-

vertical features. In this case, the value of each vertical and query-vertical feature depends on the vertical associated with the training set instance. The basic intuition behind this approach is to learn a model that focuses on portable features (equally correlated with relevance across verticals) and ignores non-portable features (unequally correlated with relevance across verticals).

One drawback of this approach is that it may focus on non-portable features that are predictive for a popular vertical with many positive instances in the joint training set. For example, a popular travel vertical may cause the model to focus on a feature that considers whether the query is travel related, which may not be predictive features for the target vertical. A second, slightly different, approach is to weight each training instance inversely proportional to the number of positive instances for that vertical in the joint training set  $Q_*$ . Arguello et al. [2010] refer to this process as *vertical balancing*.

A third approach to learning a portable model is to train a model using only the subset of features more likely to be portable, which can be automatically identified in advance. Arguello et al. [2010] identified the most portable features by treating each feature as single-evidence predictor and measuring the harmonic mean of prediction performance across all source verticals. As one might expect, for example, a feature that considers the query-likelihood score given the vertical’s query-log language model performs consistently better for different verticals than a feature that considers whether the query is travel-related.

Arguello et al. [2010] present an evaluation where both vertical-balancing and portable feature selection improved vertical selection performance for the target vertical.

**Model Adaptability.** Prior work on vertical selection found that, for some verticals, non-portable features are highly predictive. For example, Arguello et al. [2009b] report on a feature ablation analysis where query-category features were amongst the most highly predictive. Query-category features are non-portable because different verticals are likely to focus on different topical domains. Arguello et al. [2010] also experimented with an approach for exploiting vertical-specific, non-portable evidence uniquely predictive for the target vertical.

The proposed approach builds on a domain adaptation approach referred to as Tree-based Domain Adaptation (TRADA) [Chen et al., 2011]. The basic idea is to train a model using training data from one or more source domains, and then to continue the training phase using a small amount of target-domain training data. Arguello et al. [2010] proposed a solution that proceeds in four steps: (1) learn a portable model using source vertical training data, (2) make predictions for a set of queries with respect to the target vertical, (3) treat the most confident predictions as true positive and negative target-vertical examples for training, and (4) use the TRADA approach to re-tune the model's parameters given this small amount of target-vertical training data.

The adaptable model outperformed all three portable models described above.

## 6.2 Smoothing Vertical Click Data

A search engine can use click data to improve its ranking function. Suppose, for example, that a user *skips* document  $d_1$  and *clicks* on document  $d_2$ . From the system's perspective, this is evidence that  $d_2$  should have been ranked above  $d_1$  for this query. If there are many such cases, then the system might decide to *re-consider* how it is combining the different sources of evidence being used to score documents in response to a query. Radlinski and Joachims [2005] describe an approach for training a RankSVM learning-to-rank model using click information.

In the context of aggregated search, vertical clicks are very sparse because they are *heavily* skewed towards the top few vertical results that are displayed in an aggregated search. Seo et al. [2011] proposed a method for smoothing vertical result clicks by diffusing click information from the top-ranked vertical results (displayed on the aggregated SERP) to lower-ranked results (not displayed on the aggregated SERP) based on their similarity. In information retrieval, the *cluster hypothesis* states that similar documents (in a text-similarity sense) are relevant to the same information requests [Rijsbergen, 1979]. The approach from Seo et al. [2011] is motivated by the idea that similar documents should

have similar click counts for the same query.<sup>1</sup>

The vertical click-smoothing approach from Seo et al. [2011] proceeds in two steps. First, given a query where vertical  $v$  is displayed on the SERP and one of its top- $t_v$  aggregated results is clicked, it computes the text-based similarity between each clicked result and each non-clicked result in the top  $T$  (where  $T > t_v$ ). Second, the approach diffuses part of this click to other documents based on their similarity. Clicks were only diffused to other high quality results. The approach focused on smoothing vertical clicks for a community question answering (CQA) vertical, and therefore answer quality was estimated using the technique from Jeon et al. [2006]. Seo et al. [2011] found that training a learning-to-rank (LTR) model using *diffused* click data improved retrieval performance.

### 6.3 Composite Retrieval

The goal of *composite retrieval* is to organize search results into *bundles* associated with different *aspects* or *sub-topics* of the query. For examples, a travel-related query may return bundles associated with travel options, accommodations, local events, restaurants, and points of interest. Several studies have considered composite retrieval within the context of aggregated search, where the bundles may combine results from different sources (i.e., web results and results from different verticals) [Bota et al., 2014, 2015].

Bota et al. [2015] report on a user study where participants were given general search tasks (e.g., “Write a blog post about living in India”) and were asked to organize relevant documents from potentially different sources into bundles reflecting different aspects of the information-seeking task. Participants were not given any guidance about what bundles to create for each task. However, they were asked to name their bundles in order to determine whether different participants created similar bundles for the same task. Study results found four major trends. First, at least for the tasks used in this study,

---

<sup>1</sup>In a similar way, Diaz [2007] proposed an approach for retrieval performance prediction that considers the extent to which similar documents obtain a similar retrieval score.

there was reasonable agreement between the bundles created by different participants for the same task. Specifically, for 85% of the tasks used, half of the participants created bundles with *at least* two common aspects or sub-topics. Second, there was a tendency to associate certain documents with multiple bundles. These were referred to as *pivot* documents. Results found that most pivot documents were web documents rather than vertical documents. This lends support to the current aggregated search paradigm of always displaying web results on the SERP. Third, about 80% of all bundles had results from more than one source, suggesting that vertical results were also useful. Finally, participants were also asked to rate bundles in terms of four criteria (relevance, diversity, cohesion, and freshness). Results found that relevance, cohesion, and diversity were the most important criteria for determining the overall quality of a bundle. However, agreement between participants across these three criteria was fairly low, suggesting that bundle quality is highly subjective.

Together, these results suggest that certain tasks are likely to be associated with at least some subset of commonly agreed-upon aspects or sub-topics that a system could return as bundles of results (e.g., “planning a trip”). Moreover, results suggest that users are likely to gain from seeing bundles that include vertical documents in addition to web documents.

Bota et al. [2014] proposed several algorithms for constructing bundles using results originating from different sources (i.e., web and vertical results). The proposed algorithms aimed to produce bundles with four different criteria: (1) the bundles contain documents that are topically relevant to the task (*relevance*), (2) each bundle represents a coherent aspect of the task (*topical cohesion*), (3) different bundles represent different aspects of the task (*topical diversity*), and (4) the bundles contain relevant results from different sources (*vertical diversity*). Topical cohesion was operationalized using the average similarity between document-pairs in the *same* bundle (higher is better), and topical diversity was operationalized using the average similarity between document-pairs in *different* bundles (lower is better).

Composite retrieval is an interesting new search paradigm and sev-

eral open questions remain for future work. First, it is unclear how the system should present bundles to users and how users interact with bundled results. For example, are users able to easily understand the different aspects represented in different bundles? Prior work on automatically labeling document clusters may be useful for describing bundles to users [Treeratpituk and Callan, 2006]. Second, once user behavior with composite retrieval interfaces is better understood, future work will likely develop new evaluation methodologies and metrics for composite retrieval.

## 6.4 Query Disambiguation and Vertical Selection

The goal of query disambiguation is to automatically identify the different aspects or sub-topics associated with an ambiguous, multi-faceted, or task-oriented query. For example, given the query “iphone 6”, a system might predict sub-topics such as “iphone 6 sales”, “iphone 6 review”, “iphone 6 features”, and “iphone 6 look and feel”. One interesting approach to vertical selection is to first identify the different sub-topics associated with the query, and then identify the vertical(s) that best fit each sub-topic. For example, a system might predict the following sub-topic/vertical pairs: (“iphone 6 sales”, news), (“iphone 6 review”, web), (“iphone 6 features”, web), and (“iphone 6 look and feel”, images).

This two-step approach to vertical selection was the focus of the Query Understanding Task at NTCIR 2016 Yamamoto et al. [2016]. Participating systems were given a set of ambiguous, multi-faceted, or task-oriented queries and were asked to produce a ranking of no more than 10 sub-topics per query, and for each sub-topic, a ranking of verticals from a pre-defined set (e.g., web, images, encyclopedia, news, shopping, and Q&A).

The best-performing group in the NTCIR 2016 Query Understanding Task used the following approach [Nanba et al., 2016]. The first goal was to produce a set of sub-topics. To this end, the system gathered a set of candidate sub-topics from different sources: noun phrases appearing top-ranked documents; query suggestions returned from dif-

ferent query suggestion APIs; and queries from a query-log associated with the same search session, topical category, or the same clicked documents as the original query. Then, the system clustered the candidate sub-topics into 10 clusters using the top-10 retrieved documents as each candidate sub-topic's vector representation. Finally, the system selected terms from each clusters as the sub-topic title using term frequency information.

The second goal was to rank verticals for each subtopic. The system scored verticals in response to a sub-topic using the following procedure. First, the authors manually coded a collection of webpages as being associated with a candidate vertical. Then, they trained a classifier to associate webpages with a particular vertical using a set of manually selected cue phrases as features. Finally, the system scored verticals using the top-500 results returned from this collection of vertical-classified pages. Each vertical's score was proportional to the number of top-500 documents associated with the vertical.

This is an interesting approach to aggregated search that might get more attention in future work. The general idea is to first generate a more complete representation of the different possible intents associated with the input query, and to then perform vertical selection for each intent separately.

## 6.5 Aggregated Search for Children

Young children and teenagers account for a significant portion of all search engine users. For example, one report claims that in 2013, 78.5% of children between the ages of 3-17 had access to a home computer and that 57.1% regularly used the internet [ChildTrends, 2015]. In this section, we review prior work on aggregated search for young users. Prior research in this area has focused on two main questions: (1) How are the search behaviors of children and teenagers different from those of adults? and (2) How can we tailor aggregated search solutions for children?

**Search Behavior of Children.** Duarte Torres et al. [2010] analyzed queries and search sessions from the AOL query-log that had

clicks on content intended for children 12 years old or younger. Results suggest that children issue longer queries and more natural language queries, have a flatter click distribution, and have search sessions where they issue more queries and click on more documents, possibly because they have less domain knowledge, less experience identifying relevant content, or less experience using a search interface [Bilal, 2002]. In a follow-up analysis, Duarte Torres and Weber [2011b] considered queries issued to a commercial search engine from users for which age was known. Results found that children were less likely to use search assistance (e.g., spelling correction), targeted content with a more basic reading-level, and, targeted more content associated the “games” domain. By comparison, teenagers targeted more content related to music and adults targeted more content associated with business. Finally, Duarte Torres et al. [2014] also found that children were more likely to follow-up engagement with knowledge-related content (e.g., Wikipedia) by searching, and were also more likely to engagement with multimedia content.

**Aggregated Search Solutions for Children.** Duarte Torres et al. [2013] focused on vertical selection in the children domain. The authors first constructed a test collection using verticals tailored for children, focusing mostly on educational material, games, entertainment (e.g., stories and coloring pages) and multimedia (e.g., images, movies, music, and videos). Also, for the purpose of evaluation, the authors included verticals not tailored for children. The test queries corresponded to AOL query-log queries with clicks on web domains geared towards children. Relevance judgements were gathered using crowd-sourced workers, who were asked to judge results based on relevance and appropriateness for children 7-12 years of age. Inter-annotator agreement was fairly high in terms of Fleiss Kappa ( $\kappa_f = 0.683$ ).

Using this test collection, Duarte Torres et al. [2013] proposed a variant of the ReDDE algorithm (Equation 2.4) that favors verticals with age-appropriate content. Recall that ReDDE uses a retrieval from a centralized sample index (CSI) to estimate the number of query-related documents in each vertical. Each top-ranked CSI result casts a number of “votes” in favor of its originating vertical, where the number



of votes is proportional to the estimated number of documents in the vertical. In the proposed ReDDE variant, the number of votes is proportional to *estimated* ratio of age-appropriate to non-age-appropriate documents in the vertical.

Duarte Torres et al. [2013] estimated the number of age-appropriate and non-age-appropriate documents in each vertical using the *multiple capture-recapture* method for collection size estimation [Shokouhi et al., 2006b]. This approach estimates a collection’s size by conducting multiple capture-recapture steps (Equation 2.5). First, it uses query-based sampling to gather  $k$  samples of size  $T$  from vertical  $v$ . Then, the size of vertical  $v$  is given by:

$$|v| = \frac{T(T-1)k^2}{2D},$$

where  $D$  denotes the accumulated number of duplicates between all pairs of samples. Duarte Torres et al. [2013] estimated the number of age-appropriate and non-age-appropriate documents by using two different sets of AOL query-log queries for sampling: one set of queries with clicks on age-appropriate content and one with clicks on non-age-appropriate content. Results found that this ReDDE variant outperformed the original ReDDE algorithm based on its ability to rank verticals in descending order of relevance.<sup>2</sup>

## 6.6 Aggregated Mobile Search

Search from mobile devices is becoming increasingly popular. In the U.S. alone, a multi-platform survey estimated that in 2014 mobile search accounted for 29% of all search activity, with 20% conducted via smartphones and 9% via tablets [Comscore, 2015].

Mobile search poses interesting challenges and opportunities from the perspective of aggregated search. Prior work has focused on two main research areas: (1) understanding mobile information needs and their associated search behaviors, and (2) predicting user satisfaction using implicit feedback.

---

<sup>2</sup>Vertical relevance was measured based on the relevance of the top vertical results for the query.

**Mobile Information Needs and Good Abandonment.** One of the first studies in mobile search investigated the differences between abandoned queries issued from mobile devices (e.g., smartphones) versus PC devices (i.e., laptops and desktops) [Li et al., 2009]. An abandoned query is one without any clicks on the search results or a quick reformulation. The aim of the study was to compare the rate of *good* abandonment—cases where the user was able to fulfill the information need directly from the search results without clicking. The researchers manually coded different samples of abandoned mobile and PC queries from different markets (U.S., China, and Japan). Abandoned queries were labeled in terms of their potential for *good* abandonment based on the inferred information need and the search results that were displayed to the user. Out of each sample of abandoned queries, the authors reported *good* abandonment estimates of 54.8% (U.S.), 49.8% (China) and 32.3% (Japan) for mobile queries, and 31.8% (U.S.), 23.3% (China), and 19.0% (Japan) for PC queries.

These results suggest that good abandonment is much more common in mobile versus PC search. The authors discuss two possible explanations: (1) mobile users tend to restrict their queries to those that will require less interaction, and (2) mobile information needs are initiated by contextual factors (e.g., current activity, location, time, conversations with others) that motivate more “quick answer” types of searches. This second reason is supported by a prior diary study that found that 72% of participants’ mobile information needs were motivated that contextual factors [Sohn et al., 2008].

Furthermore, Li et al. [2009] identified different task categories associated with good abandonment, including: search for local businesses, answers, definitions, images, stock quotes, product information (e.g., price), movie times, weather, and language translation. Interestingly, many of these task types are now being addressed by specialized verticals in the mobile domain.

**Predicting User Satisfaction.** Prior work in this area has mostly focused on predicting user satisfaction with vertical results (or other components) that are typically not clicked (i.e., associated with good abandonment). Lagun et al. [2014b] report on a user study that fo-

cused on understanding the differences in mobile scrolling behaviors between participants presented with relevant versus non-relevant knowledge graph results (custom designed for good abandonment). The study found several important results. First, using eye-tracking data, they found that participants' visual attention focused mostly on the top-half of the phone display, which is typically associated with one or two results due to smaller display. This suggests that predicting the amount of visual attention given to a result may be possible using scrolling data alone. Second, they found that the amount of visual attention given to an item may not predict satisfaction or good abandonment. In their study, participants gave more visual attention to non-relevant versus relevant knowledge graph displays, possibly because the relevant ones provided the needed information quickly. Finally, they found that the amount visual attention given to *lower* ranked results might help predict dissatisfaction with higher ranked results. In this case, participants gave more visual attention to lower-ranked results for non-relevant versus relevant knowledge graph results.

Kiseleva et al. [2016a] focused on characterizing and predicting good abandonment involving different components aggregated on a mobile SERP (knowledge graph displays, web results, image results, and others). The authors report on a user study where participants were given search tasks that could potentially be satisfied without clicking on the SERP. As might be expected, most instances of good abandonment were associated with knowledge graph displays, which are designed to provide answers without the need of clicking. The authors also focused on predicting queries associated with good (versus bad) abandonment using features derived from scrolling behavior, attributes of the different components on the SERP, and session data (prior to the current query). Interestingly, features generated from scrolls were found to be the most predictive. The best classifier was able to predict good abandonment with about 60% precision and 80% recall.

As mentioned in Kiseleva et al. [2016a], an important task for future work is to predict *which* page element was actually responsible for the good abandonment. Together, the results from Lagun et al. [2014b] on modeling visual attention in mobile search, and the results from

Kiseleva et al. [2016a] on predicting queries with good abandonment suggest that this certainly possible.

# 7

---

## Conclusions

---

The goal of aggregated search is to combine results from different heterogeneous search services in a single presentation. Most of the published research in aggregated search has focused on the web search domain. Commercial search portals such as Google, Bing, and Yahoo! provide access to a wide range of highly specialized search services known as verticals. Example verticals include search services that focus on a particular type of media (image and video) or a particular type of search task (search for online products or local businesses).

Aggregated search is typically decomposed into two sub-tasks: (1) predicting which verticals to present in response to a query (vertical selection) and (2) predicting where to present each selected vertical on the aggregated search results page (SERP). Vertical selection is essentially a multiclass classification task. Given a query, the system must decide which verticals to select and which verticals to suppress. Vertical presentation is a more complex task, as it requires resolving contention between the different candidate verticals and deciding how to compose the final presentation of results.

Aggregated search is related to federated search, which is a more mature subfield of information retrieval. The goal of federated search

is to combine results from different collections of textual documents into a single merged ranking. Aggregated search techniques build upon decades of federated search research. However, as underscored several times in this review, aggregated search requires unique solutions. For example, aggregated search requires techniques that can combine multiple sources of evidence to make vertical selection and presentation decisions. Also, aggregated search requires techniques that can exploit a vertical-specific relation between certain sources of evidence and the relevance of a particular vertical.

In this survey, we have focused almost entirely on the web search domain. However, the algorithms, evaluation methodologies, and user studies covered in this review may have relevance to other information retrieval domains. At the core, aggregated search is driven by a “divide and conquer” approach to information retrieval. The basic approach is to develop specialized solutions for different types of content and/or different types of search tasks, and to use aggregated search techniques to provide integrated search across these different systems. As mentioned in Chapter 1, other information retrieval tasks that may benefit from the “divide and conquer” approach afforded by aggregated search include desktop search, news aggregation, contextual suggestion, and combining updates from heterogeneous social networks.

Next, we summarize some of the major trends found in prior aggregated search. Then, we conclude with a description of some potential areas for future work.

**Sources of Evidence.** The most successful approaches for vertical selection and presentation combine different sources of evidence to make predictions. Evidence can be derived from the query, the vertical, and the query-vertical pair. A type of query feature might consider the general topic of the query, a type of vertical feature might consider the number of queries recently issued to the vertical by users, and a type of query-vertical feature might consider the click-through rate for the vertical in response the same query or similar queries.

In general, pre-retrieval features, which do not require issuing the full query to the vertical search engine, are more appropriate for vertical selection. In contrast, post-retrieval features are more appropriate for

vertical presentation.

**Vertical Selection and Presentation.** The most successful approaches for vertical selection and presentation use machine learning to combine different sources of evidence as input features to a model. Using machine learning for the purpose of vertical selection and presentation poses two main changes. First, not every feature will be available for every vertical. For example, verticals that are not clickable will not have click-through features. Second, certain features will be positively correlated with relevance for some verticals, and negatively correlated for others. For example, a feature that describes whether the query is health-related will be positively correlated with relevance for a health-related vertical, but negatively correlated for a vertical that focuses on a different domain.

Approaches for vertical selection tend to use independent binary classifiers (one per vertical). In this respect, each classifier can then adopt its own feature representation and focus on the features that are *uniquely* predictive for the corresponding vertical. In other words, each vertical selection model can learn a *vertical-specific* relationship between feature values and the vertical's relevance.

Approaches for vertical presentation are more varied. Some approaches also use independent vertical-specific classifiers. Other approaches learn to predict the relative relevance between pairs of vertical and web blocks. Finally, prior work has also investigated learning-to-rank solutions. In this last case, however, the feature representation may need to be augmented to allow the model to exploit different types of evidence for different block-types.

Training a machine learning model requires training data, either derived from human-judgements or implicit feedback. In an operational setting, the system can use vertical clicks and skips to improve its performance. However, gathering feedback from the only the current model's predictions is suboptimal. For example, the vertical selection system may learn to correct false positive vertical selection predictions, but not false negative predictions. Explore/exploit methods are designed to make occasional random predictions in order to gather useful feedback. The better explore/exploit methods are strategic about when

to output a random prediction and when to output the current model's most confidence prediction.

**Evaluation.** Vertical selection is essentially a multiclass classification task. In this respect, vertical selection evaluation is fairly straight forward. Given vertical relevance judgements for a set of queries, we can evaluate a system using metrics such as accuracy, precision, recall, and f-measure.

End-to-end aggregated search evaluation is a research area in its own right. In this review, we focused mostly on test collection evaluation and online evaluation. Both methodologies have pros and cons. Test collections are portable and allow us to conduct multiple rounds of testing at little extra cost. However, relevance judgements are often made by assessors outside the context of an actual search. In a similar way, online evaluation has the benefit of using real users in real situations. As such, the system can use information about the user's preferences and the user's current context to make predictions. However, implicit feedback such as clicks and skips can be weak signals of user satisfaction.

**Studies of Search Behavior.** Behavioral studies within aggregated search have focused on two main questions: (1) What do users want from an aggregated search system? and (2) How do factors of the user, the system, the search task, and the search context influence search behaviors and outcomes with aggregated search interfaces?

The current aggregated search paradigm is to construct the aggregated SERP dynamically for each query. In this respect, the system attempts to display only the relevant verticals and presents the most relevant verticals in a more salient way. Moreover, results from the same vertical are presented together in a vertical block. With respect to the first question (What do users want?), several studies lend to support to the current paradigm. Studies have shown benefits from including vertical results alongside the web results. For example, users interact more with the vertical results when they are blended into the web results [Sushmita et al., 2009] and seeing blended vertical results can raise awareness of the contents in each vertical, which can be useful for future searches [Bron et al., 2013].



With respect to the second question, studies have found that different factors of the user, the search task, and the interface can influence search behavior. For example, one study found that users' perceptual speed can influence their performance when using an aggregated search interface [Turpin et al., 2016]. Other studies found that users interact more with content from different sources when complete more complex search tasks [Arguello et al., 2012; Jansen et al., 2009]. Finally, studies have found that results from one source on the SERP (a particular vertical) can influence user engagement with the results from other sources (the web results) [Arguello and Capra, 2016].

## 7.1 Future Directions

Next, we discuss some possible directions for future work.

**Harnessing Session-based Evidence.** Users often conduct multiple searches as they attempt to satisfy an information need. In the context of ad-hoc retrieval, session-based evidence, derived from previous user interactions within the same search session, can help improve retrieval. In fact, this was the main goal of the TREC Session Track, which ran from 2010 to 2014. Specifically, the goal was to use session information such the previous queries, clicks, and dwell-times in order to improve results for the current query. In the 2014, almost all participating systems were able to use session data to improve retrieval performance for the current query Carterette et al. [2014].

Prior work in aggregated search has not considered session-based evidence for improving vertical selection and presentation decisions. None of the features evaluated in prior work derive evidence from the current search session. For example, we could imagine that clicks on certain web results may suggest that a particular vertical is relevant. For example, a click on an online retailer may suggest that the user has shopping vertical intent. Session-based evidence may improve vertical selection and presentation performance.

**Personalized Aggregated Search.** Current aggregated search systems provide a “one size fits all” solution for users. However, users may have different search styles, preferences, cognitive abilities, and

mental models about the system. Prior work on ad-hoc retrieval found that search experience and perceptual speed (a type of cognitive ability) had an effect on search effectiveness [Al-Maskari and Sanderson, 2011]. In the context of aggregated search, Turpin et al. [2016] also found that participants with high perceptual speed rated their experience with an aggregated search interface as superior than participants with low perceptual speed.

Future research in aggregated search might need to consider individual user characteristics in deciding how to aggregate results. Work towards this goal requires addressing three main questions. First, we need to determine which characteristics of a user have the greatest influence on search behavior and performance with aggregated search systems. Second, we need to understand how to better present results to individuals with certain characteristics. Finally, we need to develop models that can *infer* these characteristics for real users based on their interactions with the system.

**Wholepage Presentation.** Current aggregated search systems make two sets of predictions: predicting which verticals to display and where to display each selected vertical. Some of the end-to-end systems covered in this review assume that verticals are presented by simply displaying the top  $t_v$  results returned by the vertical in response to the query. In this respect, the system must simply decide which verticals to display and where to display each selected vertical block. These layout constraints could be greatly relaxed. For example, the system could predict not only which verticals to display and where, but also: (1) which results from a particular vertical to display, (2) how many results to display, and (3) how they should be displayed on the SERP. For example, Arguello and Capra [2016] found that when image vertical results were displayed to the right side of the web results and with a border and different-colored background, the image results had no influence on user interaction with other results.

The main challenge in developing an aggregated search framework with fewer constraints is evaluation. To date, there is no evaluation metric that can reliably measure wholepage quality

**Aggregated Search in New Environments.** Information re-

trieval systems are constantly evolving and exploring new frontiers. Aggregated search technologies may play an important role in advancing two areas of future research: (1) dialogue-based search and (2) intelligent agents in collaborative environments.

Intelligent personal assistants such as Apple’s Siri, Google Now, and Microsoft’s Cortana can respond to user’ spoken information requests using either spoken answers and/or search results. Current systems can even preserve context across queries. For example, a user can issue the request “current weather in chapel hill” followed by “how about tomorrow?”. The most recently published research in the area has focused on three main areas: (1) understanding how spoken information requests differ from textual queries [Guy, 2016], (2) understanding what types of interactions lead to user satisfaction and incorporating these into methods of automatic evaluation [Jiang et al., 2015; Kiseleva et al., 2016b,a], and (3) developing techniques for effectively communicating search results using a speech-only communication channel [Trippas, 2015].

The ultimate goal in this vein of research is to have a dialogue-based system that can, not only preserve context, but also ask follow-up questions to disambiguate the user’s information need and/or to understand the user’s current context in order to retrieve the most relevant information or to communicate the information found in the most appropriate way. Trippas [2015] argues that a necessary step towards this goal is a tighter integration between the search system and the conversational agent.

It seems unlikely that a single system will be able to engage in human-like dialogue for all different types of search tasks, such as search for local venues, products, weather, quick answers, and for more exploratory information-seeking tasks. In this respect, aggregated search technologies may provide an alternative based on its signature “divide and conquer” approach. We may be able to develop specialized conversational agents for different task types, domains, and back-end systems, and combine them using a centralized broker that directs users to the most relevant conversational agent. At the start of the dialogue and as it progresses, the system would need to predict which dialogue agent

may be better able to assist the user.

The second area where aggregated search may play an important role is on research in intelligent agents in collaborative environments. Chat applications such as Facebook messenger, Slack, and Skype messenger allow people to communicate and collaborate on shared tasks. More and more, chat applications provide access to agents, or *chat bots*, that users can interact with to perform certain tasks. For example, Skype provides chat bots that can respond to search for specific types of content (e.g., news, images, videos, music, memes), summarize web-pages in response a URL, and provide travel information. Individual users and groups can request information from a chat bot by *explicitly* sending requests to the bot and responding to follow-up questions (if any). Chat bots that provide search results typically provide a summary of the search results directly in the chat channel and links for the user(s) to explore the results in a browser window.

Future research might consider developing search-based chat bots that intervene in conversations where they can assist by providing relevant information. This is off course, a challenging task. A search bot would need to intervene at the appropriate time and, possibly, after having learned about the information need or task from the ongoing conversation. Prior work points to several different reasons for why users do not engage with help systems, including the cost of cognitively disengaging from the main task, and the fear of unproductive help-seeking [Dworman and Rosenbaum, 2004].

Aggregated search techniques may be useful in predicting which chat bot might be able to help in a particular conversation (if any). However, several challenges need to be addressed: (1) predicting which chat bot is relevant (if any), (2) mining the conversation for information about the current task in order to intervene with some level of prior knowledge, and (3) intervening at a point in which the users are likely to engage with the assistance.

## References

---

- Abou-Assaleh, T. and Gao, W. Geographic ranking for a local search engine. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 911–911, New York, NY, USA, 2007. ACM.
- Agrawal, R., Gollapudi, S., Halverson, A., and Ieong, S. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, New York, NY, USA, 2009. ACM.
- Al-Maskari, A. and Sanderson, M. The effect of user characteristics on search effectiveness in information retrieval. *Inf. Process. Manage.*, 47(5):719–729, September 2011.
- Arguello, J. Improving aggregated search coherence. In Hanbury, A., Kazai, G., Rauber, A., and Fuhr, N., editors, *Advances in Information Retrieval*, volume 9022 of *Lecture Notes in Computer Science*, pages 25–36. Springer International Publishing, 2015.
- Arguello, J. and Capra, R. The effects of aggregated search coherence on search behavior. *ACM Transactions on Information Systems*, 11(1), 2016.
- Arguello, J. and Capra, R. The effect of aggregated search coherence on search behavior. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1293–1302, New York, NY, USA, 2012. ACM.

- Arguello, J. and Capra, R. The effects of vertical rank and border on aggregated search coherence and search behavior. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14*, pages 539–548, New York, NY, USA, 2014. ACM.
- Arguello, J., Callan, J., and Diaz, F. Classification-based resource selection. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, pages 1277–1286, New York, NY, USA, 2009a. ACM.
- Arguello, J., Diaz, F., Callan, J., and Crespo, J.-F. Sources of evidence for vertical selection. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, pages 315–322, New York, NY, USA, 2009b. ACM.
- Arguello, J., Diaz, F., and Paiement, J.-F. Vertical selection in the presence of unlabeled verticals. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10*, pages 691–698, New York, NY, USA, 2010. ACM.
- Arguello, J., Diaz, F., and Callan, J. Learning to aggregate vertical results into web search results. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 201–210, New York, NY, USA, 2011a. ACM.
- Arguello, J., Diaz, F., Callan, J., and Carterette, B. A methodology for evaluating aggregated search results. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval, ECIR '11*, pages 141–152, Berlin, Heidelberg, 2011b. Springer-Verlag.
- Arguello, J., Wu, W.-C., Kelly, D., and Edwards, A. Task complexity, vertical display, and user interaction in aggregated search. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, pages 435–444, New York, NY, USA, 2012. ACM.
- Arguello, J., Capra, R., and Wu, W.-C. Factors affecting aggregated search coherence and search behavior. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management, CIKM '13*, pages 1989–1998, New York, NY, USA, 2013. ACM.
- Aula, A., Khan, R. M., and Guan, Z. How does search behavior change as search becomes more difficult? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10*, pages 35–44, New York, NY, USA, 2010. ACM.

- Bailey, P., Craswell, N., White, R. W., Chen, L., Satyanarayana, A., and Tahaghoghi, S. M. Evaluating search systems using result page context. In *Proceedings of the Third Symposium on Information Interaction in Context, IiX '10*, pages 105–114, New York, NY, USA, 2010a. ACM.
- Bailey, P., Craswell, N., White, R. W., Chen, L., Satyanarayana, A., and Tahaghoghi, S. Evaluating whole-page relevance. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10*, pages 767–768, New York, NY, USA, 2010b. ACM.
- Bennett, P. N., Svore, K., and Dumais, S. T. Classification-enhanced ranking. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 111–120, New York, NY, USA, 2010. ACM.
- Beverly, R. and Afergan, M. Machine learning for efficient neighbor selection in unstructured p2p networks. In *Proceedings of the 2Nd USENIX Workshop on Tackling Computer Systems Problems with Machine Learning Techniques, SYSML'07*, pages 1:1–1:6, Berkeley, CA, USA, 2007. USENIX Association.
- Bharat, K., Kamba, T., and Albers, M. Personalized, interactive news on the web. *Multimedia Systems*, 6(5):349–358, 1998.
- Bian, J., Chang, Y., Fu, Y., and Chen, W.-Y. Learning to blend vitality rankings from heterogeneous social networks. *Neurocomputing*, 97:390 – 397, 2012. ISSN 0925-2312.
- Bilal, D. Children’s use of the yahooligans! web search engine. iii. cognitive and physical behaviors on fully self-generated search tasks. *Journal of the American Society for Information Science and Technology*, 53(13):1170–1183, 2002.
- Bota, H., Zhou, K., Jose, J. M., and Lalmas, M. Composite retrieval of heterogeneous web search. In *Proceedings of the 23rd International Conference on World Wide Web, WWW '14*, pages 119–130, New York, NY, USA, 2014. ACM.
- Bota, H., Zhou, K., and Jose, J. J. *Advances in Information Retrieval: 37th European Conference on IR Research, ECIR 2015, Vienna, Austria, March 29 - April 2, 2015. Proceedings*, chapter Exploring Composite Retrieval from the Users’ Perspective, pages 13–24. Springer International Publishing, 2015.
- Bota, H., Zhou, K., and Jose, J. M. Playing your cards right: The effect of entity cards on search behaviour and workload. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval, CHIIR '16*, pages 131–140, New York, NY, USA, 2016. ACM.

- Brennan, K., Kelly, D., and Arguello, J. The effect of cognitive abilities on information search for tasks of varying levels of complexity. In *Proceedings of the 5th Information Interaction in Context Symposium, IiX '14*, pages 165–174, New York, NY, USA, 2014. ACM.
- Broder, A. A taxonomy of web search. *SIGIR Forum*, 36(2), September 2002.
- Broder, A., Fontoura, M., Josifovski, V., and Riedel, L. A semantic approach to contextual advertising. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07*, pages 559–566, New York, NY, USA, 2007. ACM.
- Bron, M., van Gorp, J., Nack, F., Baltussen, L. B., and de Rijke, M. Aggregated search interface preferences in multi-session search tasks. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13*, pages 123–132, New York, NY, USA, 2013. ACM.
- Callan, J. and Connell, M. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19:97–130, 2001a.
- Callan, J. and Connell, M. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19:97–130, 2001b.
- Capra, R., Arguello, J., and Scholer, F. Augmenting web search surrogates with images. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM '13*, pages 399–408, New York, NY, USA, 2013. ACM.
- Carterette, B., Kanoulas, E., Hall, M., and Clough, P. Overview of the trec 2014 session track. In *Proceedings of the 24th Text Retrieval Conference, TREC '14*. NIST, 2014.
- Caverlee, J., Liu, L., and Bae, J. Distributed query sampling: A quality-conscious approach. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '06*, pages 340–347, New York, NY, USA, 2006. ACM.
- Chapelle, O., Joachims, T., Radlinski, F., and Yue, Y. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions of Information Systems*, 30(1):6:1–6:41, 2012.
- Chen, D., Chen, W., Wang, H., Chen, Z., and Yang, Q. Beyond ten blue links: Enabling user click modeling in federated web search. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12*, pages 463–472, New York, NY, USA, 2012. ACM.
- Chen, K., Bai, J., and Zheng, Z. Ranking function adaptation with boosting trees. *ACM Trans. Inf. Syst.*, 29(4):18:1–18:31, December 2011.



- Chen, Y., Liu, Y., Zhou, K., Wang, M., Zhang, M., and Ma, S. Does vertical bring more satisfaction?: Predicting search satisfaction in a heterogeneous environment. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1581–1590, New York, NY, USA, 2015. ACM.
- ChildTrends. Home computer access and internet use. <http://www.childtrends.org/?indicators=home-computer-access>, 2015. Accessed: 2016-05-31.
- Chuklin, A., Schuth, A., Hofmann, K., Serdyukov, P., and de Rijke, M. Evaluating aggregated search using interleaving. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 669–678, New York, NY, USA, 2013a. ACM.
- Chuklin, A., Serdyukov, P., and de Rijke, M. Click model-based information retrieval metrics. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 493–502, New York, NY, USA, 2013b. ACM.
- Cleverdon, C. W. The aslib cranfield research project on the comparative efficiency of indexing systems. *Aslib Proceedings*, 12(12):421–431, 1960.
- Comscore. Digital Future in Focus U.S. 2015. Technical report, 2015.
- Cronen-Townsend, S., Zhou, Y., and Croft, W. B. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 299–306, New York, NY, USA, 2002. ACM.
- Dean-Hall, A., Clarke, C. L. A., Kamps, J., Thomas, P., and Voorhees, E. Overview of the trec 2012 contextual suggestion track. In *Proceedings of the 21st Text Retrieval Conference*, TREC '12. NIST, 2012.
- Dean-Hall, A., Clarke, C. L. A., Simone, N., Kamps, J., Thomas, P., and Voorhees, E. Overview of the trec 2013 contextual suggestion track. In *Proceedings of the 22nd Text Retrieval Conference*, TREC '13. NIST, 2013.
- Dean-Hall, A., Clarke, C. L. A., Kamps, J., Thomas, P., and Voorhees, E. Overview of the trec 2014 contextual suggestion track. In *Proceedings of the 23rd Text Retrieval Conference*, TREC '14. NIST, 2014.
- Dean-Hall, A., Clarke, C. L. A., Kamps, J., Kiseleva, J., and Voorhees, E. Overview of the trec 2014 contextual suggestion track. In *Proceedings of the 24th Text Retrieval Conference*, TREC '15. NIST, 2015.
- Demeester, T., Trieschnigg, D., Nguyen, D., and Hiemstra, D. Overview of the trec 2013 federated web search track. In *Proceedings of the 23rd Text Retrieval Conference*, TREC '13. NIST, 2013.

- Demeester, T., Trieschnigg, D., Nguyen, D., Hiemstra, D., and Zhou, K. Overview of the trec 2014 federated web search track. In *Proceedings of the 23rd Text Retrieval Conference, TREC '14*. NIST, 2014.
- Diaz, F. Performance prediction using spatial autocorrelation. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07*, pages 583–590, New York, NY, USA, 2007. ACM.
- Diaz, F. Integration of news content into web results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining, WSDM '09*, pages 182–191, New York, NY, USA, 2009. ACM.
- Diaz, F. and Arguello, J. Adaptation of offline vertical selection predictions in the presence of user feedback. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, pages 323–330, New York, NY, USA, 2009. ACM.
- Diaz, F., White, R., Buscher, G., and Liebling, D. Robust models of mouse movement on dynamic web search results pages. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management, CIKM '13*, pages 1451–1460, New York, NY, USA, 2013. ACM. URL <http://doi.acm.org/10.1145/2505515.2505717>.
- Duarte Torres, S. and Weber, I. What and how children search on the web. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 393–402, New York, NY, USA, 2011a. ACM.
- Duarte Torres, S. and Weber, I. What and how children search on the web. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 393–402, New York, NY, USA, 2011b.
- Duarte Torres, S., Hiemstra, D., and Serdyukov, P. Query log analysis in the context of information retrieval for children. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10*, pages 847–848, New York, NY, USA, 2010. ACM.
- Duarte Torres, S., Hiemstra, D., and Huibers, T. Vertical selection in the information domain of children. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '13*, pages 57–66, New York, NY, USA, 2013. ACM.
- Duarte Torres, S., Weber, I., and Hiemstra, D. Analysis of search and browsing behavior of young users on the web. *ACM Trans. Web*, (2):7:1–7:54, 2014.

- Dumais, S., Cutrell, E., and Chen, H. Optimizing search by showing results in context. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '01, pages 277–284, New York, NY, USA, 2001. ACM.
- Dworman, G. and Rosenbaum, S. Helping users to use help: Improving interaction with help systems. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1717–1718, New York, NY, USA, 2004. ACM.
- Ekstrom, R., French, J., Harman, H., and Dermen, D. *Kit of Factor-Referenced Cognitive Tests*. Educational Testing Service, Princeton, NJ, USA, 1979.
- Feng, Y. and Lapata, M. Topic models for image annotation and text illustration. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 831–839, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- Fleiss, J. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- Friedman, J. H. Stochastic gradient boosting. *Comput. Stat. Data Anal.*, 38(4):367–378, February 2002.
- Gravano, L., Chang, C.-C. K., García-Molina, H., and Paepcke, A. Starts: Stanford proposal for internet meta-searching. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, SIGMOD '97, pages 207–218, New York, NY, USA, 1997. ACM.
- Guy, I. Searching by talking: Analysis of voice queries on mobile web search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 35–44, New York, NY, USA, 2016. ACM.
- Hassan, A., White, R. W., Dumais, S. T., and Wang, Y.-M. Struggling or exploring?: Disambiguating long search sessions. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 53–62, New York, NY, USA, 2014. ACM.
2010. Hauff, C. *Predicting the Effectiveness of Queries and Retrieval Systems*. dissertation, Univeristy of Twente, 2010.
- Hofmann, K., Whiteson, S., and de Rijke, M. A probabilistic method for inferring preferences from clicks. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 249–258, New York, NY, USA, 2011. ACM.

- Hofmann, K., Behr, F., and Radlinski, F. On caption bias in interleaving experiments. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 115–124, New York, NY, USA, 2012. ACM.
- Hofmann, K., Whiteson, S., and Rijke, M. D. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions of Information Systems*, 31(4):17:1–17:43, November 2013.
- Hong, D., Si, L., Bracke, P., Witt, M., and Juchcinski, T. A joint probabilistic classification model for resource selection. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10*, pages 98–105, New York, NY, USA, 2010. ACM.
- Hong, L., Dom, B., Gurumurthy, S., and Tsioutsoulouklis, K. A time-dependent topic model for multiple text streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11*, pages 832–840, New York, NY, USA, 2011. ACM.
- Huang, J., White, R. W., and Dumais, S. No clicks, no problem: Using cursor movements to understand and improve search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11*, pages 1225–1234, New York, NY, USA, 2011. ACM.
- Huang, J., White, R., and Buscher, G. User see, user point: Gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 1341–1350, New York, NY, USA, 2012a. ACM.
- Huang, J., White, R. W., Buscher, G., and Wang, K. Improving searcher models using mouse cursor activity. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, pages 195–204, New York, NY, USA, 2012b. ACM.
- Jansen, B. J., Booth, D. L., and Spink, A. Determining the informational, navigational, and transactional intent of web queries. *Inf. Process. Manage.*, 44(3):1251–1266, May 2008.
- Jansen, B. J., Booth, D., and Smith, B. Using the taxonomy of cognitive learning to model online searching. *Inf. Process. Manage.*, 45(6):643–663, November 2009.
- Järvelin, K. and Kekäläinen, J. Cumulated gain-based evaluation of ir techniques. *ACM Transactions of Information Systems*, 20(4):422–446, 2002.

- Jeon, J., Croft, W. B., Lee, J. H., and Park, S. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 228–235, New York, NY, USA, 2006. ACM.
- Jiang, J., Hassan Awadallah, A., Jones, R., Ozertem, U., Zitouni, I., Gurunath Kulkarni, R., and Khan, O. Z. Automatic online evaluation of intelligent assistants. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, New York, NY, USA, 2015. ACM.
- Jie, L., Lamkhede, S., Sapra, R., Hsu, E., Song, H., and Chang, Y. A unified search federation system based on online user feedback. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1195–1203, New York, NY, USA, 2013. ACM.
- Joachims, T. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM.
- Katja Hofmann, F. R., Lihong Li. Online evaluation for information retrieval. *Foundations and Trends(R) in Information Retrieval*, 10:1–117, June 2016.
- Khelghati, M., Hiemstra, D., and van Keulen, M. Size estimation of non-cooperative data collections. In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, IIWAS '12, pages 239–246, New York, NY, USA, 2012. ACM.
- Kim, J. and Croft, W. B. Ranking using multiple document types in desktop search. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 50–57, New York, NY, USA, 2010. ACM.
- Kiseleva, J., Williams, K., Hassan Awadallah, A., Crook, A. C., Zitouni, I., and Anastasakos, T. Predicting user satisfaction with intelligent assistants. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 45–54, New York, NY, USA, 2016a. ACM.
- Kiseleva, J., Williams, K., Jiang, J., Hassan Awadallah, A., Crook, A. C., Zitouni, I., and Anastasakos, T. Understanding user satisfaction with intelligent assistants. In *Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 121–130, New York, NY, USA, 2016b. ACM.
- Koffka, K. *Principles of Gestalt psychology*. Harcourt, New York, 1935.

- König, A. C., Gamon, M., and Wu, Q. Click-through prediction for news queries. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 347–354, New York, NY, USA, 2009. ACM.
- Koolen, M., Kazai, G., and Craswell, N. Wikipedia pages as entry points for book search. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 44–53, New York, NY, USA, 2009. ACM.
- Krakovsky, M. All the news that's fit for you. *Communications of the ACM*, 54(6):20–21, 2011.
- Kulkarni, A. and Callan, J. Selective search: Efficient and effective search of large textual collections. *ACM Transactions on Information Systems*, 33(4):17:1–17:33, 2015.
- Kumar, R. and Vassilvitskii, S. Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 571–580, New York, NY, USA, 2010. ACM.
- Lagun, D. and Agichtein, E. Inferring searcher attention by jointly modeling user interactions and content salience. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 483–492, New York, NY, USA, 2015. ACM.
- Lagun, D., Ageev, M., Guo, Q., and Agichtein, E. Discovering common motifs in cursor movement data for improving web search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 183–192, New York, NY, USA, 2014a. ACM. ISBN 978-1-4503-2351-2.
- Lagun, D., Hsieh, C.-H., Webster, D., and Navalpakkam, V. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 113–122, New York, NY, USA, 2014b. ACM. ISBN 978-1-4503-2257-7.
- Lavrenko, V. and Croft, W. B. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.
- Lawrence, S., Bollacker, K., and Giles, C. L. Indexing and retrieval of scientific literature. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, pages 139–146, New York, NY, USA, 1999. ACM.

- Lee, C.-J., Croft, W. B., and Kim, J. Y. Evaluating search in personal social media collections. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, WSDM '12*, pages 683–692, New York, NY, USA, 2012. ACM.
- Li, J., Huffman, S., and Tokuda, A. Good abandonment in mobile and pc internet search. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09*, pages 43–50, New York, NY, USA, 2009. ACM.
- Li, X., Wang, Y.-Y., and Acero, A. Learning query intent from regularized click graphs. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08*, pages 339–346, New York, NY, USA, 2008. ACM.
- Liu, J., Cole, M. J., Liu, C., Bierig, R., Gwizdka, J., Belkin, N. J., Zhang, J., and Zhang, X. Search behaviors in different task types. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries, JCDL '10*, pages 69–78, New York, NY, USA, 2010a. ACM.
- Liu, J., Liu, C., Gwizdka, J., and Belkin, N. J. Can search systems detect users' task difficulty?: Some behavioral signals. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10*, pages 845–846, New York, NY, USA, 2010b. ACM.
- Liu, J., Liu, C., Cole, M., Belkin, N. J., and Zhang, X. Exploring and predicting search task difficulty. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 1313–1322, New York, NY, USA, 2012. ACM.
- Liu, K.-L., Santoso, A., Yu, C., and Meng, W. Discovering the representative of a search engine. In *Proceedings of the Tenth International Conference on Information and Knowledge Management, CIKM '01*, pages 577–579, New York, NY, USA, 2001. ACM.
- Liu, K.-L., Yu, C., and Meng, W. Discovering the representative of a search engine. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM '02*, pages 652–654, New York, NY, USA, 2002. ACM.
- Liu, Y., Liu, Z., Zhou, K., Wang, M., Luan, H., Wang, C., Zhang, M., and Ma, S. Predicting search user examination with visual saliency. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '16*, pages 619–628, New York, NY, USA, 2016. ACM.

- Liu, Z., Liu, Y., Zhou, K., Zhang, M., and Ma, S. Influence of vertical result in web search examination. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 193–202, New York, NY, USA, 2015. ACM.
- Long, B. and Chang, Y. *Relevance Ranking for Vertical Search Engines*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2014.
2007. Lu, J. *Full-text Federated Search in Peer-to-peer Networks*. PhD thesis, Pittsburgh, PA, USA, 2007.
- Luo, C., Liu, Y., Zhang, M., and Ma, S. *Query Ambiguity Identification Based on User Behavior Information*, pages 36–47. AIRS 2014. Springer International Publishing, 2014.
- Lv, Y., Moon, T., Kolari, P., Zheng, Z., Wang, X., and Chang, Y. Learning to model relatedness for news recommendation. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 57–66, New York, NY, USA, 2011. ACM.
- Markov, I., Kharitonov, E., Nikulin, V., Serdyukov, P., de Rijke, M., and Crestani, F. Vertical-aware click model-based effectiveness metrics. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 1867–1870, New York, NY, USA, 2014. ACM.
- McCreadie, R. and Macdonald, C. Relevance in microblogs: Enhancing tweet retrieval using hyperlinked documents. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, OAIR '13, pages 189–196, Paris, France, France, 2013.
- Metzler, D., Dumais, S., and Meek, C. Similarity measures for short segments of text. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 16–27, Berlin, Heidelberg, 2007. Springer-Verlag.
- Moffat, A. and Zobel, J. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions of Information Systems*, 27(1):2:1–2:27, 2008.
- Nanba, H., Sakai, T., Kando, N., ana Koji Eguchi, A. K., Hatano, K., Shimizu, T., Hirate, Y., and Fujii, A. Nexti at ntcir-12 imine-2 task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, NTCIR '16. National Institute of Informatics, 2016.



- Navalpakkam, V., Jentzsch, L., Sayres, R., Ravi, S., Ahmed, A., and Smola, A. Measurement and modeling of eye-mouse behavior in the presence of nonlinear page layouts. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13*, pages 953–964, New York, NY, USA, 2013. ACM.
- Nygren, E. Between the clicks: Skilled users scanning of pages. In *Proceedings of Designing for the Web: Empirical Studies*, 1996.
- Palmer, S. E. Common region: A new principle of perceptual grouping. *Cognitive Psychology*, 24(3):436 – 447, 1992.
- Ponnuswami, A. K., Pattabiraman, K., Brand, D., and Kanungo, T. Model characterization curves for federated search using click-logs: Predicting user engagement metrics for the span of feasible operating points. In *Proceedings of the 20th International Conference on World Wide Web, WWW '11*, pages 67–76, New York, NY, USA, 2011a. ACM.
- Ponnuswami, A. K., Pattabiraman, K., Wu, Q., Gilad-Bachrach, R., and Kanungo, T. On composition of a federated web search result page: Using online users to provide pairwise preference for heterogeneous verticals. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11*, pages 715–724, New York, NY, USA, 2011b. ACM.
- Radlinski, F. and Joachims, T. Query chains: Learning to rank from implicit feedback. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD '05*, pages 239–248, New York, NY, USA, 2005. ACM.
- Radlinski, F., Kurup, M., and Joachims, T. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08*, pages 43–52, New York, NY, USA, 2008. ACM.
- Rijsbergen, C. J. V. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.
- Sahami, M. and Heilman, T. D. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the 15th International Conference on World Wide Web, WWW '06*, pages 377–386, New York, NY, USA, 2006. ACM.
- Sanderson, M. Ambiguous queries: Test collections need more sense. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08*, pages 499–506, New York, NY, USA, 2008. ACM.

- Santos, R. L., Macdonald, C., and Ounis, I. Learning to rank query suggestions for adhoc and diversity search. *Inf. Retr.*, 16(4):429–451, 2013.
- Santos, R. L. T., Macdonald, C., and Ounis, I. Aggregated search result diversification. In *Proceedings of the Third International Conference on Advances in Information Retrieval Theory*, ICTIR'11, pages 250–261, Berlin, Heidelberg, 2011. Springer-Verlag.
- Schulze, M. A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method. *Social Choice and Welfare*, 36(2):267–303, 2011.
- Seo, J., Croft, W. B., Kim, K. H., and Lee, J. H. Smoothing click counts for aggregated vertical search. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval*, ECIR'11, pages 387–398, Berlin, Heidelberg, 2011. Springer-Verlag.
- Shen, D., Pan, R., Sun, J.-T., Pan, J. J., Wu, K., Yin, J., and Yang, Q. Query enrichment for web-query classification. *ACM Trans. Inf. Syst.*, 24(3):320–352, 2006.
- Shokouhi, M. Central-rank-based collection selection in uncooperative distributed information retrieval. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 160–172, Berlin, Heidelberg, 2007. Springer-Verlag.
- Shokouhi, M. Learning to personalize query auto-completion. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 103–112, New York, NY, USA, 2013. ACM.
- Shokouhi, M. and Si, L. Federated search. *Found. Trends Inf. Retr.*, 5(1): 1–102, January 2011.
- Shokouhi, M., Scholer, F., and Zobel, J. Sample sizes for query probing in uncooperative distributed information retrieval. In *Proceedings of the 8th Asia-Pacific Web Conference on Frontiers of WWW Research and Development*, APWeb'06, pages 63–75, Berlin, Heidelberg, 2006a. Springer-Verlag.
- Shokouhi, M., Zobel, J., Scholer, F., and Tahaghoghi, S. M. M. Capturing collection size for distributed non-cooperative retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 316–323, New York, NY, USA, 2006b. ACM.

- Si, L. and Callan, J. Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 298–305, New York, NY, USA, 2003a. ACM. ISBN 1-58113-646-3.
- Si, L. and Callan, J. A semisupervised learning method to merge search engine results. *ACM Transactions on Information Systems*, 21(4):457–491, 2003b.
- Si, L., Jin, R., Callan, J., and Ogilvie, P. A language modeling framework for resource selection and results merging. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 391–397, New York, NY, USA, 2002. ACM. ISBN 1-58113-492-4.
- Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 254–263, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
- Sohn, T., Li, K. A., Griswold, W. G., and Hollan, J. D. A diary study of mobile information needs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 433–442, New York, NY, USA, 2008. ACM.
- Sushmita, S., Joho, H., and Lalmas, M. A task-based evaluation of an aggregated search interface. In *Proceedings of the 16th International Symposium on String Processing and Information Retrieval*, SPIRE '09, pages 322–333, Berlin, Heidelberg, 2009. Springer-Verlag.
- Sushmita, S., Joho, H., Lalmas, M., and Jose, J. M. Understanding domain relevance in web search. In *WWW Workshop on Web Search Result Summarization and Presentation*, 2010a.
- Sushmita, S., Joho, H., Lalmas, M., and Villa, R. Factors affecting click-through behavior in aggregated search interfaces. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 519–528, New York, NY, USA, 2010b. ACM. URL <http://doi.acm.org/10.1145/1871437.1871506>.
- Sushmita, S., Piwowarski, B., and Lalmas, M. Dynamics of genre and domain intents. In *Proceedings of the 6th Asia Information Retrieval Societies Conference*, AAIRS '10, pages 399–409, Berlin, Heidelberg, 2010c. Springer-Verlag.
- Sutton, R. S. and Barto, A. G. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

- Thomas, P. and Shokouhi, M. Sushi: Scoring scaled samples for server selection. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 419–426, New York, NY, USA, 2009. ACM.
- Treeratpituk, P. and Callan, J. Automatically labeling hierarchical clusters. In *Proceedings of the 2006 International Conference on Digital Government Research*, dg.o '06, pages 167–176. Digital Government Society of North America, 2006. . URL <http://dx.doi.org/10.1145/1146598.1146650>.
- Trippas, J. R. Spoken conversational search: Information retrieval over a speech-only communication channel. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 1067–1067, New York, NY, USA, 2015. ACM.
- Tsur, G., Pinter, Y., Szpektor, I., and Carmel, D. Identifying web queries with question intent. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 783–793, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.
- Turpin, L., Kelly, D., and Arguello, J. To blend or not to blend? perceptual speed, visual memory and aggregated search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, New York, NY, USA, 2016. ACM.
- Wang, C., Liu, Y., Zhang, M., Ma, S., Zheng, M., Qian, J., and Zhang, K. Incorporating vertical results into search click models. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 503–512, New York, NY, USA, 2013. ACM.
- Wang, Y., Yin, D., Jie, L., Wang, P., Yamada, M., Chang, Y., and Mei, Q. Beyond ranking: Optimizing whole-page presentation. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 103–112, New York, NY, USA, 2016. ACM.
- Wen, J.-R., Nie, J.-Y., and Zhang, H.-J. Clustering user queries of a search engine. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 162–168, New York, NY, USA, 2001. ACM.

- Wu, W.-C., Kelly, D., Edwards, A., and Arguello, J. Grannies, tanning beds, tattoos and nascar: Evaluation of search tasks with varying levels of cognitive complexity. In *Proceedings of the 4th Information Interaction in Context Symposium, IIX '12*, pages 254–257, New York, NY, USA, 2012. ACM.
- Xu, J. and Li, H. Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '07*, pages 391–398, New York, NY, USA, 2007. ACM.
- Xue, X.-B., Zhou, Z.-H., and Zhang, Z. M. Improving web search using image snippets. *ACM Transactions of Internet Technology*, 8(4):21:1–21:28, 2008.
- Yamamoto, T., Liu, Y., Zhang, M., Dou, Z., Zhou, K., Markov, I., Kato, M. P., Ohshima, H., and Fujita, S. Overview of the ntcir-12 imine-2 task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR '16*. National Institute of Informatics, 2016.
- Yue, Y., Patel, R., and Roehrig, H. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 1011–1018, New York, NY, USA, 2010. ACM.
- Zhou, K., Cummins, R., Halvey, M., Lalmas, M., and Jose, J. M. Assessing and predicting vertical intent for web queries. In *Proceedings of the 34th European Conference on Advances in Information Retrieval, ECIR'12*, pages 499–502, Berlin, Heidelberg, 2012a. Springer-Verlag.
- Zhou, K., Cummins, R., Lalmas, M., and Jose, J. M. Evaluating reward and risk for vertical selection. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM '12*, pages 2631–2634, New York, NY, USA, 2012b. ACM. ISBN 978-1-4503-1156-4.
- Zhou, K., Cummins, R., Lalmas, M., and Jose, J. M. Evaluating aggregated search pages. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, pages 115–124, New York, NY, USA, 2012c. ACM. URL <http://doi.acm.org/10.1145/2348283.2348302>.
- Zhou, K., Cummins, R., Lalmas, M., and Jose, J. M. Which vertical search engines are relevant? In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13*, pages 1557–1568, New York, NY, USA, 2013a. ACM.

- Zhou, K., Lalmas, M., Sakai, T., Cummins, R., and Jose, J. M. On the reliability and intuitiveness of aggregated search metrics. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management, CIKM '13*, pages 689–698, New York, NY, USA, 2013b. ACM.
- Zhou, K., Demeester, T., Nguyen, D., Hiemstra, D., and Trieschnigg, D. Aligning vertical collection relevance with user intent. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14*, pages 1915–1918, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1.
- Zhuang, J., Mei, T., Hoi, S. C., Xu, Y.-Q., and Li, S. When recommendation meets mobile: Contextual and personalized recommendation on the go. In *Proceedings of the 13th International Conference on Ubiquitous Computing, UbiComp '11*, pages 153–162, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0630-0.