Universitat Pompeu Fabra

Master Thesis

Analyzing clickthrough data to evaluate Freesound retrieval quality

Author:
Dara Dabiri

Supervisor:
Dr. Xavier Serra, Frederic Font

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in the

Music Technology Group
Department de Tecnologies de la Informació i les Communicacions

s

Declaration of Authorship

I, Dara Dabiri, declare that this thesis titled, 'Analyzing clickthrough data to evaluate Freesound retrieval quality' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"It takes a good crisis to get us going. When we feel fear and we fear loss we are capable of quite extraordinary things."

Paul Gilding

UNIVERSITAT POMPEU FABRA

Abstract

Faculty Name
Department de Tecnologies de la Informació i les Communicacions

Master of Science

Analyzing clickthrough data to evaluate Freesound retrieval quality

by Dara Dabiri

Freesound.org is an online collaborative sound database where people from different disciplines share recorded sound clips under Creative Commons licenses. Freesound's search functionality is one the feature that is used by thousands of users everyday. Due to the rapid expansion of the sound collection and the variety within sounds, ranking quality of search results has become an important part of the platform. Automatically judging the quality of the ranking algorithms based on user clickthrough data holds promise for analyzing retrieval quality faster and cheaper. We investigate whether certain observable statistics relate to retrieval quality of Freesound. We propose a model that leverages the thousands of clicks received by Freesound to predict the retrieval quality. Six absolute metrics as usage statistics are hypothesized to motonically change with retrieval quality. We design three ranking strategies and expose users to each of these rankings as the default sorting of search results. We show that the change in metrics is correlated with the rank correlation of different rankings.

Acknowledgements

I would like to thank my supervisors Xavier Serra and Frederic Font for giving me the opportunity to work on the Freesound Project. Their feedback was instrumental in reaching the finish line and their advice was essential in maintaining the research direction.

Throughout my internship at Freesound, I truly enjoyed collaborating with Gerard Roma who was always available to answer my technical questions. His knowledge of Freesound gave me ample opportunities to experiment with different ways of implementing my ideas.

I would also like to extend my gratitude to Alastair Porter for providing the right tools and resources for collecting clicthrough data from Freesound. I also enjoyed receiving his invaluable feedback during code reviews and assessment of my ideas.

I am indepted to my professors from UPF whose teachings fed my thirst for knowledge music, sound and technology and their evaluation was an important part of my development.

I owe it to Laia for her constant support and encouragement. She played a crucial role in helping me keep a postivie attitude and overcome the obstacles.

Last but not leaset, I am grateful to my family. Without their support none of this was possible. They have been a constant source of inspiration in my life and my thoughts and decisions have always been influenced by their guidance and encouragements.

Contents

List of Figures

List of Tables

To my grandfathers, Parviz Dabiri and Hussein Meymandi

Chapter 1

Introduction

Freesound is an online community of sound practitioners. It has an active userbase who produce, share and download content at a high volume on a daily basis. Freesound's large and diverse database of sounds has led many sound consumers to use Freesound as their chosen sound catalogue. This has resulted in a need to provide the latest technology in information retrieval and search functionality with an specific outlook on sound and musical content.

Originally, Freesound sorted search results based on popularity of the sound. Sounds that have a full match between the query and their textual data are ordered based on the number of downloads prior to that search. One of the main fundamental challenges within Freesound is ranking sounds with respect to user's queries. Information about each sound is gathered from the uploader in form of tags, sound description and title. We need to know how much weight we should put on each of these fields when computing a relevancy of a sound to the query. Are tags more infomrative than the title of the sound? Should sounds be ordered based on their popularity or should they be ordered by the distance between the textual content and the query?

We first give an overview of the use of clickthrough data in the field of information retrieval and search engines. We breifly discuss advantages of using click data in modelling user behavior, automatic annotation of documents, machine-learnt ranking, query expansion and retrieval quality. We conclude the state of the art by shining some lights on the use of clickthrough data in sound information retrieval and their potential in improving retrieval quality in the case of Freesound.

In this study we investigate whether click data has any significant information to improve Freesound search engine. Specifically, we want to know how we can use such a data source to evaluate the quality of ranking sounds in reponse to user query. We mainly

focus on the queries that users submit, the sounds that the search engine lists as the search results and the sounds users preview and/or download from those results. By studying users' implicit interaction we propose some absolute metrics that may change as response to change of ranking quality. Using absolute metrics for evaluation follows the hypothesis that ranking quality influencess observable user behavior in an absolute sense (e.g. better retrieval leads to higher-ranked clicks, better retrieval leads to faster clicks).

Chapter 2

State Of The Art

The sheer size of online multimedia on the internet has led to the special focus within the information retrieval community to devise optimized strategies in indexing and retrieval of online multimedia. The cheap access to memory, interconnectivity and popularity of user generated content has become the central elements in this expansion of multimedia content. On the other hand, search engines are the primary tools for locating content and finding answers to information needs. Additionally, the field of multimedia information retrieval deals with extracting, summarizing information and semantics that is both explicitly and implicitly contained in multimedia data. Compared to traditional text mining, multimedia data mining is a much more challenging process [1]. First, the data is inherently unstructured meaning there are no well-defined fields that categorize each document at different abstraction levels. Second, it is highly heterogeneous meaning its digitization process undergoes a variety of different transformation steps and it exists in many different formats and standards. Third, it is highly multidimensional which makes the feature extraction and storage process very expensive and time consuming. Lastly, the interpretation of multimedia data is very subjective. For example, two people could describe the same medium in two completely different ways.

2.1 Challenges In Multimedia Information Retrieval

In [2], Ponceleon and Slaney group these challenges into three categories: the semantic gap, feature ambiguity and machine-generated data. The semantic gap is the term used to describe the large gap between the machine readable description of a multimedia signal and the actual interpretation of the data by a human. Among all the media types on the web, text has the smallest semantic gap due to availability of structure. On the other hand, music poses the largest semantic gap due to high level of complexity and

subjectivity in the content. Feature ambiguity refers to generation of multiple interpretation of a single data source based on which features are selected in the classification process. Mood detection in music data mining can provides a good example of how different feature selection and parametrization can result in two different interpretation on a single song. For example, within a small analysis frame, the algorithm could classify a song as sad while whithin a larger frame it would classify the same song as happy. Machine generated data refers to the scope at which growth of data must be matched by robust algorithms that can automatically generate semantic information. Every second, terabytes of multimedia content are being uploaded to the web and it requires fast and comprehensive algorithms that can mine important data for later retrieval.

2.2 Common Solutions

Due to incomplete content descriptions and feature extraction algorithms, many systems have traditionally relied on expert annotation as the main strategy in filling the semantic gap. A popular example is Getty images [1] : a website licensing the use of professionally authored images. The content of this portal is tagged and annotated by the editorial staff. Other strategies have applied crowd annotation based on the notion of wisdom of the crowd. Delicious [2] uses personal bookmarks of its users as a source for annotating links and web pages. Similarly, gamification of the annotation process provides a collaborative environment between players to annotate images in a competitive setting [3]. All of the these strategies are still largely expensive. Recently, researchers have looked at user (implicit) interaction with search engine as a source of information for data mining. A large volume of data can be collected at very little cost (and merely no cost at user's end). There is an implicit cognitive process that happens which can be mined for semantic feature extraction or retrieval optimization. In this study, we aim to use information from these interaction to automatically gather semantic information about contents of a sound sharing website and to understand the community of searchers.

2.3 Clickthrough Data

Clickthrough data and the click data mining in general are one of the central frontiers in information retrieval research and commercial search engines [3]. They have been studied and employed from a variety of different angles with the purpose of improving search engines. Use of click data in modern information retrieval can be grouped into 4

[1] www.gettyimages.com
[2] www.delicious.com
[3] semanticgames.org

categories: 1) Usage behavior modelling, 2) Document annotation, 3) Ranking and 4) Query expansion/suggestion. In the following sections, we will review these categories in two contexts: i) text retrieval, ii) multimedia retrieval. We will show that usage of clickthrough data is also starting to become a frontier in multimedia information retrieval research. Sound information retrieval is affected by the same challenges. To the best of our knowledge, no comprehensive work has been done in this field in the context of general sound information retrieval. Here by sound information retrieval (SIR), we refer to a retrieval system that indexes and searches through heterogeneous audio content from musical performance and synthetic sounds to environmental and Foley recordings. Most of the work within the community has been focused on music information retrieval and speech recognition. Although there have been some groundbreaking developments in both content and context analysis, the algorithms developed in one field (e.g. music data mining) are sometimes difficult to apply another field (e.g. sound effects data mining).

2.3.1 Usage Behavior Modelling

By exploring usage behavior patterns, search engines can have a better understanding of the type of users , the information needs dynamics and the satisfaction level according to their services.

Baeza et al. [4] analyze query logs of a commercial search engine to model how users search and interact with search results. The authors' main idea is to demonstrate the users-search engines interaction as a 'bidirectional feedback' relation. Intuitively, the quality of search results is partially dependent on the search engine's understanding of its users and their searching process. This study tries to analyze the interaction information considering query keywords, clicks, ranking and times involved in a query session request. In a similar study, Jansen and Spink [5] investigated search trends emerging through time. They used clickthrough data logs from two different dates one year apart to isolate trends in searching and page views. They were able to demonstrate that searching behaviors evolve in certain directions with search topics becoming more diversified and interest in some categories increasing while decreasing in others.

In addition to how people search and what they search for, researchers have tried to answer the question of why people search. Understanding searcher's goal has received significant attention within the information retrieval community. In an early study, Rose and Levinson [6] manually classify user's goals to three categories: navigational, informational and resource seeking. Navigational queries aim to find specifically known websites that the user already had in mind. Informational queries aim to add to user's

knowledge by reading or viewing web pages. Resource seeking queries aim to obtain a resource (not information) available. They argue that knowing the context out of which the user formulates her query can assist the search engine in displaying or ranking the results tailored to that context. Lee et al. [7] extended user goal identification by automatically identifying goals using user-click behavior. The study follows the intuition that if a query is navigational, user will primarily click on the result she had in mind. In other words, navigational queries lead to lower click rates on higher-ranked results. Similarly, for an informational query users will click on multiple results returned by the search engine with more biased towards lower-ranked results. In another related study [8], clickthrough information was solely employed to identify users' goals. After extracting two features from the number of clicks for a query and the number of URL's clicked per query they propose a decision tree based classification algorithm to identify user goals. [9] use supervised and unsupervised machine learning algorithms to classify queries as either informational, not informational and ambiguous. Their training data consists of queries having clicks in their answers. They represent these queries as a vector of terms that appear in the documents that were clicked on as a result of searching for the query.

2.3.2 Document Annotation

Document annotation is essential to success of retrieval system. At the early stages, explicit annotations were employed. This process usually consists of editorial experts annotating data to be indexed. Considering the magnitude of user generated data, explicit annotation is almost impossible. Researchers have been looking at user implicit actions to annotate data automatically.

Xue et al. [10] investigate several strategies in generation of *surrogate documents* using data extracted from clickthrough data logs. A naive method associates query terms with clicked web pages as the metadata of web pages. The second method, co-visited method, takes into account the similarity between two pages that were answers to a common query. They list three main challenges with respect to these methods: 1) Noise in clickthrough data can introduce inaccurate metadata to associated web pages; 2) Since web users are more likely to click on a handful of top results, they clickthrough data is very sparse and 3) New web pages receive no clicks since they rarely occur within the top ranked results. To elevate these challenges, a third method, iterative reinforcement algorithm, considers similarity between two pages as a metric for determining the usability of their shared queries. If the similarity is above a certain threshold, the queries associated with one page are assigned to other similar pages. This method is performed exhaustively until the algorithm reaches a fixed point. The results showed significant

improvement in retrieval quality of all three methods over the baseline with the iterative reinforcement algorithm having the best performance.

Image retrieval systems have paid special attention to automatic annotation and usage behavior as sources of semantic tagging of images on the web. These collections are highly unstructured and mostly generated by users with inadequate tag or description associated with the images. On the other hand, millions of users search for images everyday and data can be extracted from these images. In a recent survey of characteristic settings that facilitate automatic image annotations, Sawant et al. [11] enlist usage statistics computed over multiple independent users as a major source for emergence of patterns and semantics in image retrieval systems. They consider these strategies as replacements of expert annotations with crowd-sourcing; a concept that draws many parallels with the proposition of wisdom of crowds [12]. The wisdom of the crowds states that the collective judgments of many is better than the judgments made by a single person.

Ashman, et al, [13] argue that click data extracted from interaction with image search results page is significantly more reliable since results encapsulate the entire object as opposed to partial snippets in text retrieval systems. Direct image labeling and transitive image labeling are discussed and evaluated as two consensual ways to labeling images. Direct image labeling associates search terms with images that were clicked on as results of those search terms. Transitive image labeling also takes into account the labels of HTML pages that were previously classified in the same way using clickthrough data on text-based pages. They discovered that the aggregate clickthrough data applied to images is more accurate than any of the explicit methods for labeling images. In an effort to incorporate user's implicit feedback for textual annotation of images, Ntalianis et al. [14] represent each image as a hybrid vector of textual and visual elements and an environment that links textual queries to preferred documents using click data. After a user's selection, the textual query is accumulated to the images textual representation. On the other hand, the visual content similarity is used to associate untagged images with tagged ones. This strategy combined with the automatic textual annotation allows the complete annotation of non-annotated content. Image topic modeling using clickthrough data demonstrates improvements in document similarity computation [15].

2.3.3 Ranking

Ranking is one of the most important task every search engine has to deal with. Some of the most critical challenges in ranking are the quality of ranking signals and the ranking function definition [2]. Concepts such as PageRank [16], HITS [17], etc. are different

ranking strategies search engines use to stay competitive. Click data are used as means to improve quality and also ranking signals. Ranking solutions strive to order the most relevant and quality documents with respect to user's information needs. Intuitively, clickthrough rate (CTR) could be considered as a voting measure in two different ways: 1) either the document is relevant to the user's need or 2) the quality of the clicked document is higher than the other (possibly) relevant documents. Search engines also use click data as "pseudo-votes" on new features to the search experience instead of testing users in controlled studies. In such circumstances, users have no idea their actions are being recorded by the search engine. This phenomenon is referred to as *A/B testing* within the information retrieval community [18]. In addition, some machine-learned ranking functions employ usage information as training data in the form of relevance feedback.

Researchers initially investigate relevance feedback to overcome the difficulty of formulating queries. Engaging the user in an iterative process allows the search engine and searcher to come to terms on an agreed vocabulary that best maps the query to the relevant documents [19]. The process consists of showing a set of results to the user and asking her to rate or annotate them on the degree of relevancy. Rocchio algorithm [20] is used to tune the query by maximizing similarity between query terms and terms in the relevant documents and minimizing the similarity between query terms and the terms of the non-relevant documents. However, the intrusive process of asking users to explicitly judge the relevancy of the results has shown to be a major bottleneck [21]. On the other hand, millions of users interact with search engines on a daily basis, providing valuable implicit feedback through their interaction with search results. An active area of research in information retrieval investigates automatic transformation of these implicit behaviors into relevance judgments. In an effort to employ clickthrough data as an indicator of relevance feedback, Joachims [22] introduced a learning-to-rank technique based on a pairwise approach to discriminate between relevant and non-relevant results. This approach is based on the assumption that clicked pages are more relevant than non-clicked pages skipped above. Agichtein et al. [23] suggested that aggregate clickthrough statistics can provide a more accurate preference measure compared to individual clickthrough data. They also showed how inclusion of other implicit feedback signal (e.g. time spent on results page) can improve retrieval accuracy in real-world search engines [24][25]. A comprehensive survey of implicit feedback techniques is provided by Kelly and Teevan [26].

Regarding reliability of implicit feedback interaction, evaluation of clickthrough data as an indicator of relevance judgement has been the focus of many research teams. Clickthrough data is difficult to interpret and it can be very noisy. Joachim et al. [25] discovered that clicking decisions on the results page are biased towards higher ranked

and higher quality results due to user's trust in preferred search engine and inclination towards high quality results (that may not necessarily be relevant to the information need). Eye tracking data showed that users scan the results from top to bottom most of the time. They suggested that clicked links should be assessed relative to their position among other ranked results and relative to the quality of other snippets. Agichtein et al. [23] corrected the trust bias by subtracting the background click distribution from the query's click statistics to show that the evaluated click count is a more accurate measure of relevancy. To explain the position bias further, Craswell et al. [27] describe a *cascade model* that approximates the probability of the first click when users traverse the results from top to bottom.

In the field of multimedia information retrieval, little research has been done on the reliability of clickthrough data as a measure of relevance feedback. To the best of our knowledge, the majority of this work has been done on image retrieval systems. In a recent publication, Smith et al. [28] examined the overall accuracy of image search clickthrough data. Their results showed an increased level of accuracy in clickthrough data compared to similar studies on text retreival systems. This increased accuracy can be associated with presentation of the thumbnails that encapsulate full content. Unlike text search results where a small snippet of the result is presented, image search results allow the user to completely examine them before clicking on them.

2.3.4 Query Expansion

Clickthrough data has recently become the primary source of information in another important aspect in web search: query expansion. The previous section demonstrated that the difference in domain knowledge articulation between users and search engines is one of the main challenges in information retrieval. The way a user may formulate a query may not coincide with the way documents in the corpus are annotated and the search engine cannot infer what the user is searching for. Query expansion is the process of reformulating the original query as an effort to close this gap and increase the quality of search results. Various support tools in the search process such as spell checking, stemming and synonym suggestions, are all associated with query expansion. Traditional approaches have been using document corpus to automatically build vocabularies of terms [29] [30] [31]. Recently, due to the sheer size of traffic and tremendous accumulation of query logs, Web search engines use query log corpus to map the user and document vocabulary.

The idea of using query logs has origins in the aforementioned 'wisdom of the crowds' principle which states that collective wisdom outsmarts individual expertise. Query

logs have revolutionized search support tools [18]. To elevate the difficulties in query formulation, query logs allow search engines to recommend related queries that can be selected in case of inadequate results. Another recent addition to these search support tools is query autocompletion where the system suggests completed queries before the user finishes her originally intented query. This strategy specially helps those with informational navigation needs since they are partially being educated on the most successful searches in the past. The advantage of using query logs as the source is to adapt the search to user needs and trends that change over time. However, the averaging over the whole population of searchers can possibly bury the long tail (the less popular or more specilized contents) deeper and sacrifice quality over popularity.

2.3.5 Evaluation Of Retrieval Quality

The Cranfield evaluation methodology commonly applied in TREC [**?**] uses human-adjudicated relevance . For each query, a grade assigned by an expert (e.g. Bad, Good, Excellent) specifies the relevance of each document. Given a ranking produced in response to a query, the judgments for the top ranked documents can be collected to analyze the quality of the ranking. Averaging over many queries produces average performance scores such as Normalized Discounted Cumaltive Gain (NDCG), Mean Average Precision (MAP) and Precision at K ([19]). There are some drawbacks associated with such averaging over human provided relevance judgements. For example, the expert may not be representative of all the subgroups and perspective using the retrieval algorithm. However, clickthrough data has shown to be a promising direction for evaluating retrieval quality and preference measuring.

Within this area, researchers investigate characteristics of user behavior (captured through implicit feedback measures) that can be used to evaluate ranking quality. They address the problem of predicting user search goal success by modelling their interaction with search results. The use of clickthrough data in this context allows for automatic judgement of retrieval quality. Such implicit judgements from observable behavior take advantage of clicks, query reformulations, and response times that can be collected without putting any cognitive burden on the users.

Kelly and Teevan provide an overview of some absolute metrics for evaluating ranking quality based on user behavior [26]. Ali and Chang [32] examined the relationship between relative click rates and relative human judgements of search results. Their results show that for some classes of queries higher click rate is indicative of higher relavance. Fox et al [24] used a set of queries to measure user satisfaction through analyzing implicit collected feedback. By collecting both implicit and explicit feedback

through a modified browser, they developed predictive models of user satisfaction with search results based on implicit measures.

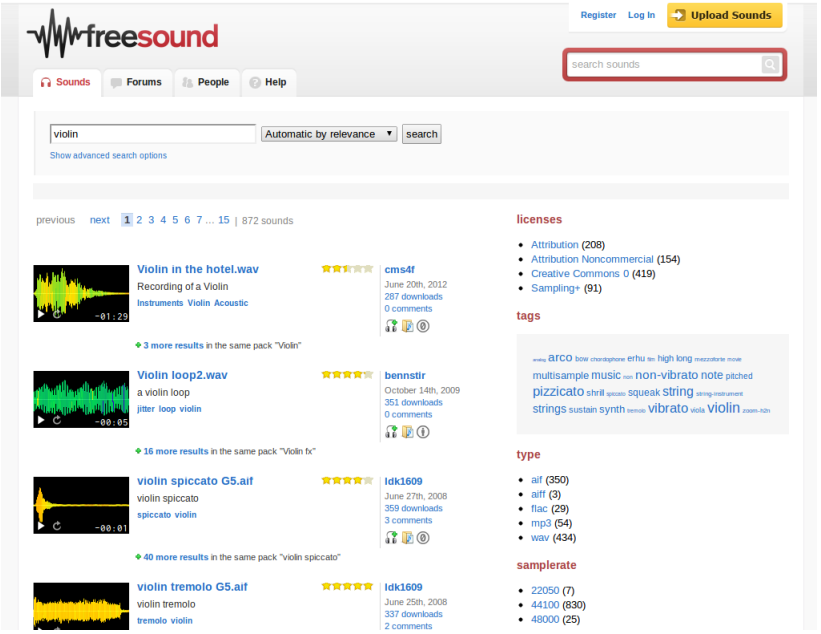

FIGURE 2.1: An example of Freesound search results page (SERP)

2.4 Clickthrough in Sound Information Retrieval

In sound information retrieval community little attention has been dedicated to the use of clickthrough data in better understanding the content and user needs. Since the inception of Naptster, the web has seen a dramatic increase in digital music and auditory content and demand for automatic understanding of audio signal by the machine is on the rise. Music Information Retrieval (MIR) is an interdisciplinary field of information retrieval scientists and signal processing experts that tries to better understand and describe sound and music signal present in the web and provide solutions to optimize retrieval. In this developing discipline, most of the focus has been on content analysis. The output of the community is mainly centered around content descriptors that take audio signal as input and produce low level descriptor that describe certain characteristics of a sound. This branch of the field is also refered to by the name of 'machine listening'. Even though significant landmarks in machine listening have been accomplished, the current state of the art descriptors are far from being commercially utilized due to lack of accuracy in their output. Therefore, most of the most prominent commercial products in soudna and music search rely on the meta data provided by the editorial staff or the user generated content. Recently, context analysis has receieved special attention mainly due to user genereated information and low-cost access to the

'wisdom of the crowd' among listeners and downloaders. On the web, portals like last.fm take into account user provided tags and playlists to better understand the musical files. Clickthough data information can be an even more resourceful signal source for better understanding the audio content and closing the semantic gap. Millions of music searchers log into sound-exclusive search engines looking for a variety of different songs and genres. Each user carries his/her own musical tastes, reasons and intent in looking for a specific medium. Clickthrough data can enrich the system by being a provider of different perspectives on a single medium. Search engines can tap into click data in search of quality of their ranking or satisfaction of their users with the search results.

The ability to preview and download provides a unique setting where the user interaction can provide valuable insight into relevancy of the sound to user's information need. For example, if a user previews and downloads a sound, the system may consider that sound more relevant to the query than the other sounds in the SERP. In terms of usage behavior modelling, there is no comprehensive study, using implicit behavior, on how and why people search in Freesound. One of the central goals of the Freesound project is social profiling and community-aware computation. Clickthrough data can provide some insights into the query space, the interaction space and the dynamics of sound information need that are specific to a sound-sharing community. Moreover, by trying to understand user's search intention, different communities of sound searchers could be served differently based on their specific needs [33]. For instance, the vocabulary of music searchers can be different from the vocabulary of field-recording searchers. On the other hand, a music searcher looking for the sound of 'violin' may be planning to use the sound for a completely different reason than a sound effects searcher. Freesound can also benefit from query expansion tools since sound articulation is a difficult task. Query expansion can elevate some of the difficulties with query formulation and educate users on related successful queries.

As the mostly widely used sound sharing website on the internet, Freesound has passed the entry barrier [18] that is required for any search engine to exploit high volume of user traffic. We believe the clickthrough information can be a tremendous resource for better understanding of sounds, users and communities.

2.4.1 SOLR Relevancy Ranking

In the following section we give a brief overview of how freesound's adapted search engine takes advantage of SOLR Relevancy functionality. Freesound uses SOLR which is an open source enterprise search platfrom from the Apache Lucene^{TM}. Relavancy is the quality of results returned in terms of which documents are found and the order that

they are presented to the user. In this document we refer to this measured quality as the SOLR scoring.

SOLR scoring uses a combination of Vector Space Model (VSM) and the Boolean model to determine how relevant a documents is to user's information need. The basic idea revolves around the fact that the more a query term occurs in a document the more relevant that document is to the query. On the other hand, VSM diminishes the effect of terms that are prevelant in large proportion of documents in the corpus.

In VSM, documents and queries are modelled as weighted vectors in a multi-dimensional space where each term is a dimension and weights are equal to $Tf - idf$ [19]:

$$tf - idf = (term\ frequency) \times (inverse\ document\ frequency) \tag{2.1}$$

Inverse document frequency of a term t in a document d is defined as follows:

$$idf_t = \log \frac{N_t}{df_t} \tag{2.2}$$

with N_t as the number of times t occurs in d and df_t as the number of times t occurs in the total collection.

The score of a document d for query q is the cosine similarity of the weighted query vectors $V(q)$ and $V(d)$.

$$cosineSimilarity(q, d) = \frac{V(q).V(d)}{|V(q)||V(d)|} \tag{2.3}$$

where $V(q).V(d)$ is the dot product of the weighted vectors and $|V(q)|$ and $|V(d)|$ are euclidean norms.

As a refinement to VSM scoring, Freesound takes advantage of two SOLR functionalities: zone scoring and boosting. Zone scoring allows the document to be broken into separate zones at index time. Freesound classifies these zones in the form of tags, descriptions and filenames of sounds. Such data is provided by the owner of the sound at upload time. Boosting allows the search engine to weight each zone differently depending on their importance. For example, if the information in the tag section is more relevant/important, a higher weight can be assigned to the tags compared to the other two sections. Freesound takes advantage of the boosting at query time. when the user posts a query to Freesound, the system assigns a set of boost values for each zone. This set are value are heuristically assigned. A major goal of this study is to investigate whether there is an optimal set of values that produced the highest level of user satisfaction.

Prior to January 2013, Freesounds default ranking of sounds was based on the number of times a sound was downloaded. In other words, once a boolean search for full text matching retrieved all the sounds containing the query terms, the results were sorted based on the number of download counts. The current version of Freesound search employs SOLR scoring as the default ranking mechanism.

Chapter 3

Methodology

3.1 Design Of The User Study

To evaluate the relationship between implicit feedback and ranking quality, we used Freesound search engine. Freesound.org is an online collaborative sound database where people from different disciplines share recorded sound clips under Creative Commons licenses. It is used daily by more than 45,000 users, predominantly sound artists and musicians for musical and post-production purposes.

The basic design of our study can be summarized into two consecutive experiments. In experiment I, starting with the original similarity-based ranking function of Freesound as the baseline model, we derive a more degraded ranking function. The degraded ranking function is designed in a way that it will always perform 'worse' than the baseline ranking function. We then expose the users of Freesound.org to these two ranking functions as detailed below and analyze whether and under which types of exposure, their observable behavior reflects the changes imposed by us in the retrieval quality.

During experiment II, we expose the users to a particular type of ordering of search results which is based on the number of times each sound is downloaded. With this option as the default sorting mechanism, Freesound collects all the sound for which a full match exists between their text and the query. Subsequently, it sorts the matched sounds based on the number of times they have been downloaded prior to the current search. With users exposed to such ranking as default, we compute the same metrics in experiment I and analyze whether any significant behavioral change is evident.

3.1.1 Construction of Ranking Functions

We start by describing how Freesound's baseline ranking algirthm scores and rank sound according to user queries. We then present our degraded ranking algorithm and explain why it performs worse than the baseline. As mentioned earlier, Freesound uses SOLR search engine. At index time 3 primary fields are selected: tags, descriptions and original filename. SOLR uses cosine similarity scoring to find and rank similar sounds according to the overlap between the query and these three fields. The baseline ranking function is favored more towards the tags. In this model the tags carry a weight of 4, descriptions carry a weight of 3 and the filenames carry a weight of 1. This is due to the fact that tags contain a more abstract information about the sound. Similarly, desciptions are more informative than filenames but usually not as well-summarized as tags. To degrade the ranking function we distributed the weights evenly between each field giving each one of them the value of 1. Table 3.1 summarizes the distribution of weights in each ranking function.

Ranking Function	Tag	Descripton	Original filename
Baseline	4	3	1
Degraded	1	1	1

TABLE 3.1: Weights assigned to each field during document scoring.

3.2 Users and System Design

Figure 3.1 illustrates the user interface of the Freesound search engine.

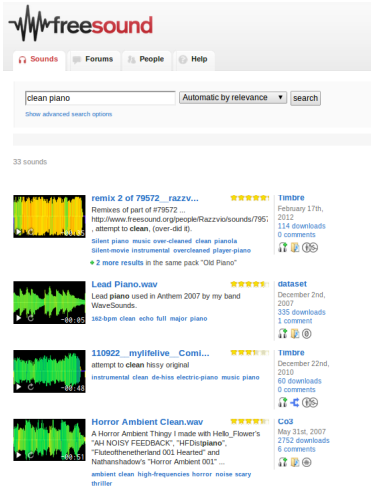

FIGURE 3.1: Screenshot of how results are presented in Freesound.

It takes a set of keywords as a query, and returns a ranking of 15 results per page. For each result, we show the waveform, original filename, a snippet of the the sound description (155 characters) and the tags associated with the sound. User has the option to preview a sound or click on the filename. Clicking on the filename will take the user to the sound page where the user can preview the sound again, read the full description or leave comments. To download a sound, the user has to have a registered Freesound account and be logged in. We register a sound preview click whenever a user clicks on the preview button layered on top of the waveform. Figures 3.3 and 3.4 show two situations where we record a preview click. When the user downloads a song we register a "sound download click".

3.2.1 User identification

Given the nature of the sound collection, consisting mostly of recordings of musical instrument and field audio and synthetic sounds and sound effects we suspect that many of our users are sound designers, music producers, muscicians and post production sound artists. On average, our search engine received about 170,000 queries per day from about 300 distinct IP ad- dresses, registering about 600000 clicks (previews + downloads) on results.

We identify each user by a unique session key that is passed to the client browser in form of a cookie. The session key is generated on default by the front-end server. Once the server assigns a session key to a user the session key inside the cookie is maintained for 14 days and renewed if the user visits Freesound after 14 days.

We define a session as a sequence of interactions (previews, downloads or queries) between a user and the search engine where less than 20 minutes pass between subsequent interactions. When associating previews and downloads to query results, we only count clicks occurring within the same session as the query. This is necessary to eliminate clicks that come from saved or cached search results or from users who were only browsing through sounds. To associate a click to a query we searched for the most recent query within the same session that had a result corresponding to the sound id of the clicked sound.

3.2.2 System Design

To establish a data collection mechanism we set up a proxy server that received logs from the main web application servers over the UDP network (Figure. 3.2). The proxy server was placed within the internal network of Universitat Pompeu Fabra that hosts

the application servers. Therefore, we speculate the rate of UDP packets being dropped during the transmission is extremely low.

Users were unaware of the changes to ranking algorithms. Integration of the logging mechanism did not affect the performance in request handling.

Upon completion of interaction log collection, we download the raw data to a machine with the following specifications: a Pentium i7 8-core processor with 6GB of RAM

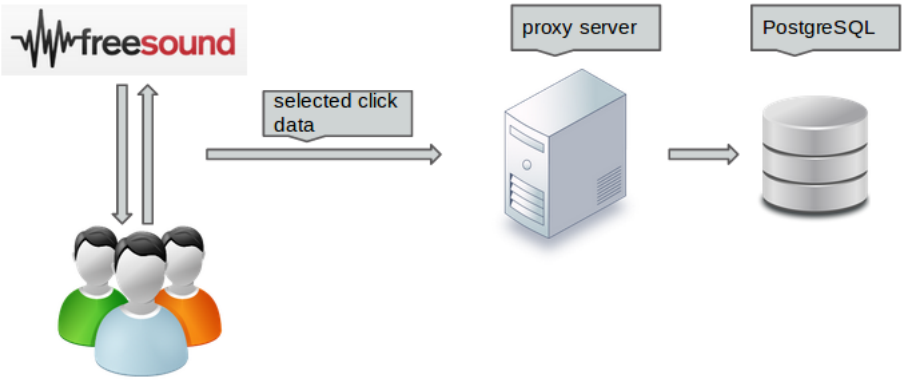

FIGURE 3.2: An overview of the data collection system

3.3 Implicit interaction and Data

Throughout this study, we recorded three types of interaction on the search results. The first interaction was recorded at the event of a preview. Users can preview sounds in to places: on the search results page (Figure 3.3) or the sound description page (Figure c). If a user wants to download a sound, they have to be logged in as a registered user. If logged in, the user can download a sound from the sound page (Figure 3.3). Everytime a user downloads the sound, we also record a 'download' interaction. Finally, with every click on the 'search' button we also record a 'query' interaction.

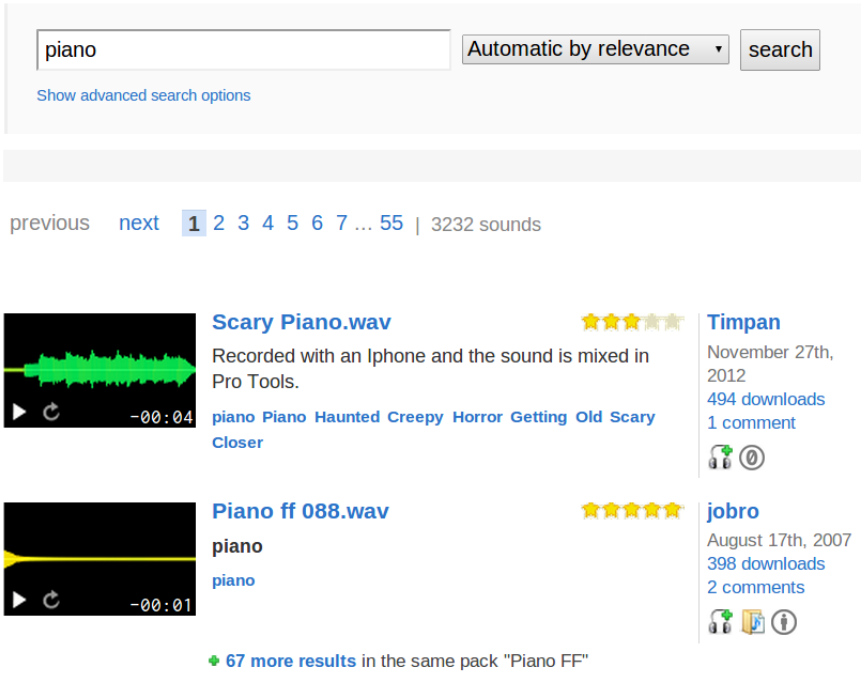

FIGURE 3.3: Screenshot of a Freesound search result page with the query 'piano' and sorting option set to Automatic by relevance (SOLR's ranking). We collected a 'preview' interaction everytime the user clicked on the play button on the lower-left corner of each sound. We also collected a query evertime the search button was clicked.

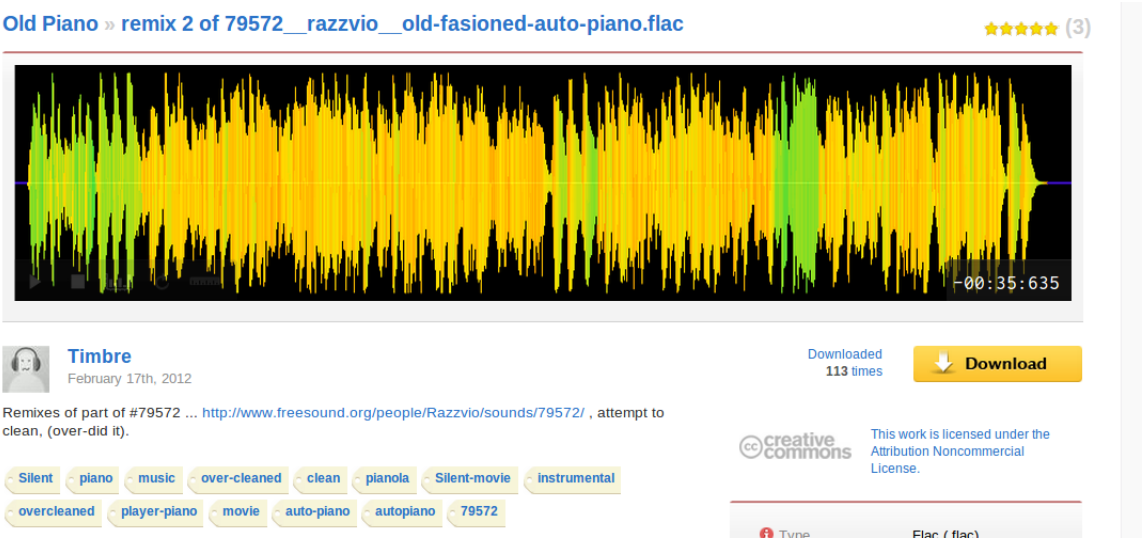

FIGURE 3.4: Screenshot of a sound page that contains the information about the sound submitted by the owner of the sound and other users' comments

3.3.1 Raw Data

In the following section, we will describe the complete process of going from the raw data to a relational database of two tables: clicks and queries.

We logged queries submitted, as well as all the preview and download clicks in the following format.

Query log:

```
[timestamp] # INFO    # QUERY : (full request) : (search-time session key)
: (sound IDs shown in the results page) : (results page number)
```

Example:

```
[2013-06-02 06:33:35,080] # INFO    # QUERY : /search/?q=wave&f=
&s=score+desc&advanced=0&g=1 : 10f93f2f59e648e9beae8e9eea446408
: [183881, 185239, 126152, 138192, 3439, 103190, 154881, 61596,
86081, 86080, 86076, 126524, 141254, 22505, 9332] : 1
```

Preview/Download Log:

```
[timestamp] # INFO    # (soundpreview|sounddownload) :
(session key of logged in users) : (session key before logging in) : (sound ID)
```

Example:

```
[2013-06-02 06:33:17,398] # INFO    # soundpreview :
3de8f9f779ad1cd1eeb72bae3bb8451b : 3de8f9f779ad1cd1eeb72bae3bb8451b : 125364
```

What follows is a detailed explanation of each component of data logs. We will review how each component was parsed and process befored being inserted in the database.

After parsing each component of each log, we placed them in two database tables. The full definition of these two database tables are provided in Appendix A.

3.3.2 Metrics

Based on the absolute metrics from [34], we propose the following metrics.

Abandonment Rate	The fraction of queries for which no results were previewed/downloaded.
Queries per session	The mean number of queries during each session
Previews per query	The mean number of preview clicks per query
Previews before download	the mean number of previews after a query and before a download
Maximum reciprocal rank	the mean value of 1/r with 'r' being the rank of the highest ranked results that is downloaded
Mean reciprocal rank	the mean of sum($1/r_i$). With r_i being the ranks of all previewed sounds that resulted in a download

Abandonment Rate. Abadonment is an important measure of user satisfaction because it shows that users had no interests in the results shown by the search engine.

Queries per session. This is a measure that indicates the rate at which users had to reformulate their queries because they could not find what they were looking for.

Previews per query. Ideally, a user wants to preview the sound to decide whether it's a sound they are looking for. The textual data might resemble some level of similairty with the query but the users tend to preview sounds before downloading. If the quality of search results is lowered in terms of relevancy to the query, users should tend to preview more sounds until they find their desired sounds.

Previews before download. This metric is very similar to the previous one with the exception that it measures whether user's will eventually download a sound. Moreover, this metric measure a type of interaction exclusive to a sound information retrieval platform. In a general purpose search engine, the actions of the user beyond looking at the snippet and clicking on a link are unkown to the server. However, here we can claim that the user received a full exposure to the content by previewing the sound before deciding to download it. If the ranking is degraded, users will tend to preview more sounds before they eventually download one.

Maximum reciprocal rank. This metric measure the highest ranked sound a user downloads. With a degraded ranking more relevant results will be pushed down the list and the metric would be decreased since it is an inverse of the highest rank.

Mean reciprocal rank. Similar to the previous metric in nature, mean reciprocal

rank measure the ranking quality of all the downloaded sound with respect to their position in the search results. Moreover, the metric should decrease as the better results are pushed down further and the top results are not as appealing.

For each metric, we hypothesize the following trends after the retrieval quality is degraded.

Metric	**Change as ranking is degraded**
Abandonment Rate	Increase (more bad results -¿ leaving sooner)
Queries per session	Increase (need to reformulate queries)
Previews per query	Increase
Previews before download	Increase
Maximum reciprocal rank	Decrease (top results are worse)
Mean reciprocal rank	Decrease (there are less relevant results)

(Appendix B contains the queries for computing each of these metrics based on the database tables defined in Appendix A)

3.3.3 Experiment Setup

We conducted two phases of clickthrough data collection each with a duration of 7 days. In phase I, we collected user interaction data based on the output of the baseline ranking function. This phase started on May 27th, 2013 and finished on June 3th, 2013. In phase II, we deployed the degraded ranking function and collected data from June 3th 2013 to June 10th, 2013. Freesound users were unaware of the changes to ranking algorithm during phase I.

Chapter 4

Results and Discussion

4.1 Results

4.1.1 Search Characteristics of Freesound Users

During the two phases of experiment I, 109430 and 106958 distinct users interacted with the search facilities of Freesound. The number was significantly lower during experiment II (Table 4.1) which is associated with the slow traffic in August.

Period	Experiment I - Phase I	Experiment I - Phase II	Experiment II
Distinct Users	109430	106958	81962

TABLE 4.1: Number of distinct users recorded in each experiment

During both phase I and phase II of the first experiment, more than 92 percent of queries requested sorting by the score (descending). Even in the second experiment where the default ranking option was set to sorting by downloads more than 57 percent of queries requested sort by score rather than sort by downloads. This shows that majority of users rely on the search engine's similarity score. Moreover, more than 98 percent of users do not use any of the advanced search option to limit the results further (Table 4.2). We can see the importance of the ranking quality and the trust users place on the freesound search engine.

advanced	count (during phase I)	count (during phase II)
off	957775	950461
on	10586	9431

TABLE 4.2: usage of advanced options to limit search results furthers

The type of queries request from Freesound range from music related terms to natural and foley sounds. The top 10 requested queries during phase I of experiment I is provided in Table 4.3.

Query	Count
explosion	5251
wind	5221
music	3699
scream	3485
piano	3349
punch	2905
rain	2758
car	2477
cartoon	2465
thunder	2441

TABLE 4.3: Top 10 queries recorded during phase I of experiment I. Phase II produced similar ordering.

We also observe that most of the interactions predominantly occur in the first page of the search results as indicated in 4.4. This is an strong indication of the value of ranking and importance of precision required to fulfill user's information needs. It also indicates the bias users will have towards the first two pages of the results. Users seldomly browse beyond the second page of the search results.

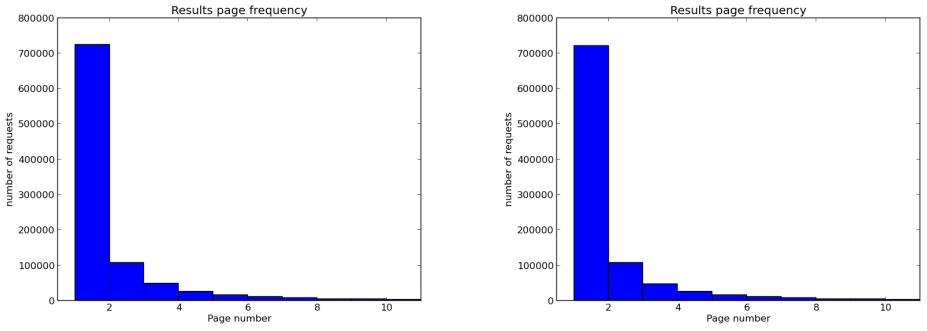

TABLE 4.4: Page number frequency for all the queries in phase I and phase II

4.1.2 Metrics

In this section we present the results of our experiments in comparing user behavior in response to changing of search results ranking. For each proposed metrics, we present a plot that shows the change in value while the number of samples increases. Each plot

consist of two curves that each corresponds to one ranking method. The measure metrics are reported in Table 4.5 for each metric corresponding to each ranking function. We have also included the hypothesized change as ranking degrades in the second column named 'Hypothesis'.

TABLE 4.5: Absolute metrics for the "Baseline vs Degraded" ranking algorithms. The second columns shows the hypothesized behavior when retrieval quality is degraded.

	Hypothesis	Baseline	Degraded
Abandonment Rate	↑	0.391	0.394
Queries per session	↑	7.04	7.04
Previews per query	↑	19.324	19.741
Previews before download	↑	4.50	4.55
Maximum reciprocal rank	↓	0.695	0.698
Mean reciprocal rank	↓	1.723	0.695

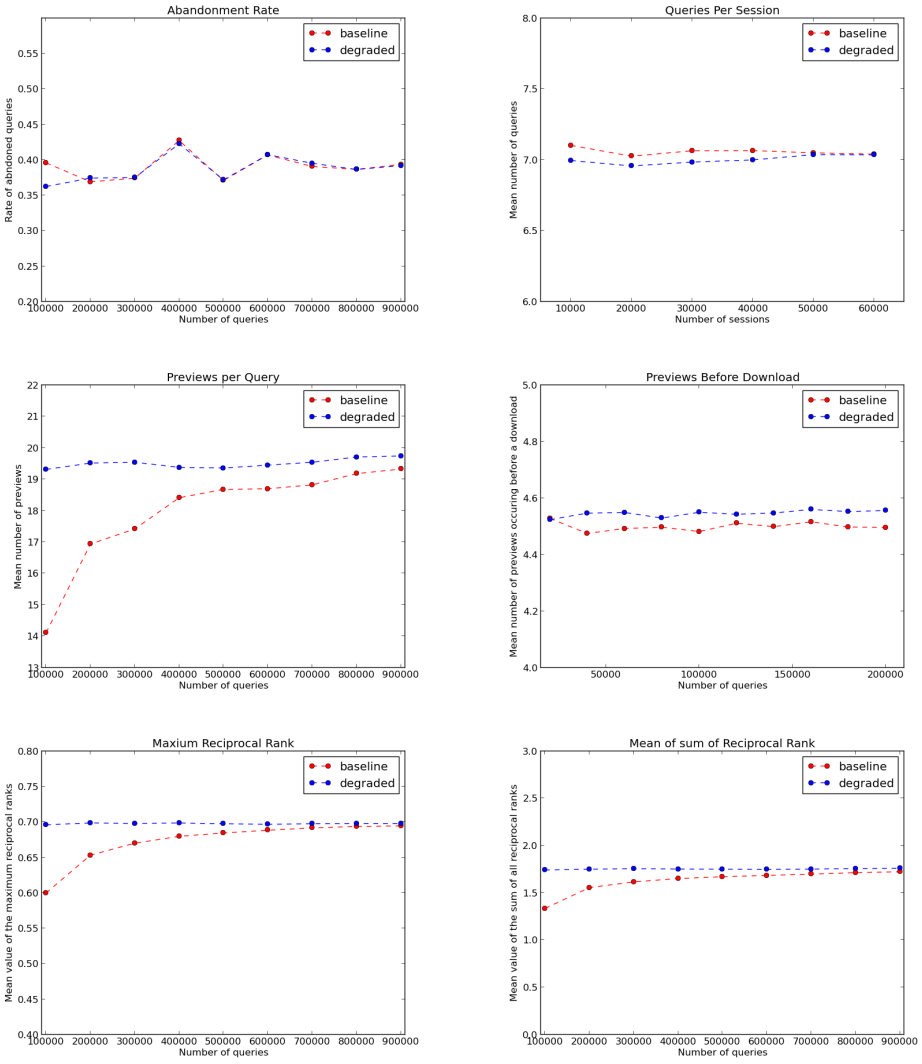

TABLE 4.6: Measurements of the six metrics for **Baseline vs. Degraded** rankings.

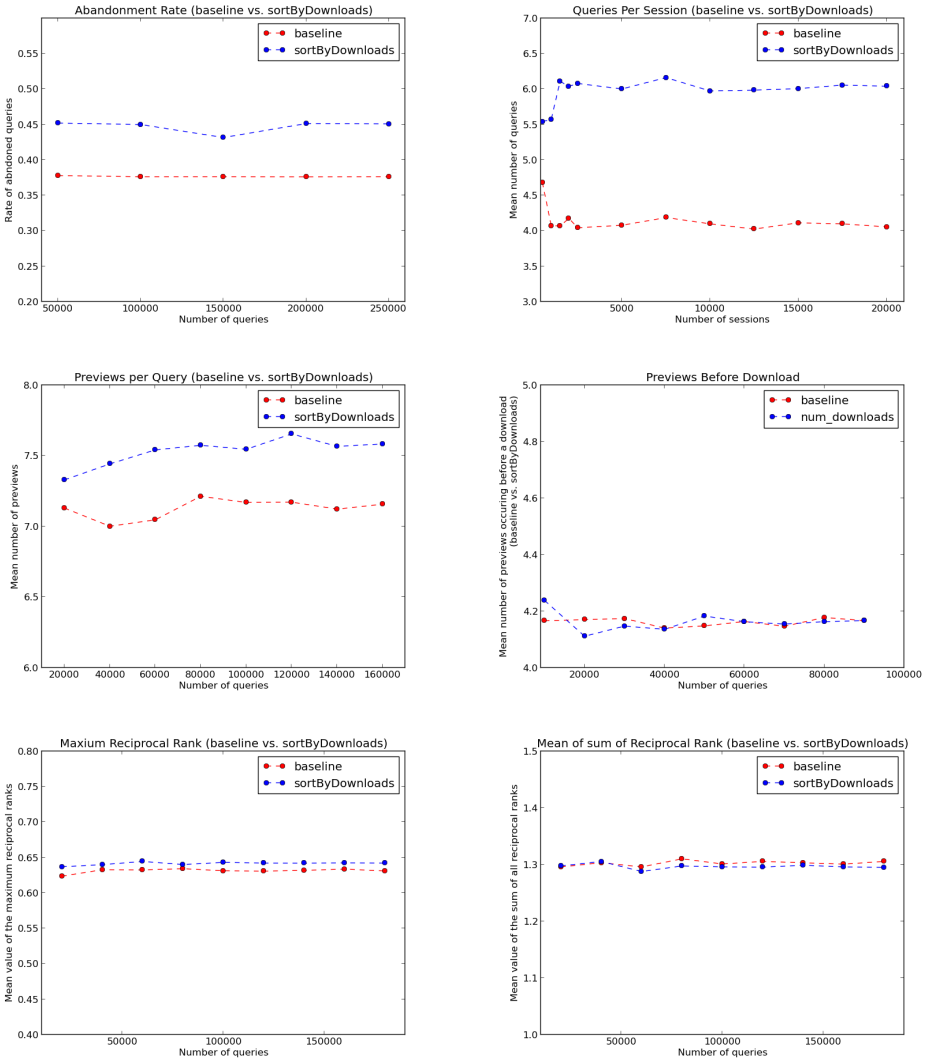

TABLE 4.7: Measurements of the six metrics for **Baseline vs. Sort By Download** rankings.

4.2 Discussion

In this section, we provide our explanation behind the different behavior of metric values in comparison of baseline ranking versus the other two rankings, namely the degraded ranking and sort by download ranking.

4.2.1 Baseline Ranking vs. Degraded Ranking

In all the plots corresponding to the metrics, as the number of samples increases the curves corresponding to each metric converge. Therefore, it is difficult to conclude that degradation of ranking has significant change in behavior measured by the proposed

metrics. In other words, the outcome of the plots neither prove nor refute the hypothesis. To explain this lack of change in behavior, we computed kendall's tau between the baseline ranking and the degraded ranking for the top 10 most frequent queries (Table 4.8). Kendall's tau [35] calculates a correlation measure for ordinal data. In other words, its output is a correspondence measure between two different ways of ordering a set. Values close to 1 suggest strong agreement between the two orderings and values close to -1 suggest strong disagreement. The computed kendall's tau for all the query terms between the baseline and degraded rankings is very close to 1. This implies that the two ordering are in very close agreement and are only slightly different. This explains why the change in behavior is not significant in any of the proposed metrics.

With respect to the maximum reciprocal rank and mean reciprocal rank, the similarity in metric value for each ranking can be associated to the users' bias towards the highest ranked results [27]. In other words, users tend to preview and eventually download the sounds that are ranked higher in the presentation order regardless of the difference in the content because of their trust in the search results ranking.

Top 10 Queries	**vs. degraded ranking**	**vs. sorting based on number of downloads**
'explosion'	0.9163530947	-0.2268597762
'wind'	0.9993196753	-0.1751230838
'music'	0.9748647422	-0.1640405592
'scream'	0.9920765133	0.1952040377
'piano'	0.9984394961	-0.2427993005
'punch'	0.9923362276	-0.0240104358
'rain'	0.9961696445	-0.2860189975
'car'	0.9981280964	-0.111264561
'cartoon'	0.999759145	-0.3401776306
'thunder'	0.9928642786	-0.3196201389

TABLE 4.8: Kendalls tau for comparing the baseline ranking vs. degraded ranking and sort by number of downloads

4.2.2 Baseline Ranking vs. Sort By Downloads Ranking

According to Table 4.8, the ordering of search results using baseline significantly disagree with the results produced using the sort by download ordering. Consequently as shown in figure 4.7 in 3 of the metrics (abandonment rate, queries per session and previews per query) there is significant change in user behavior. The distance between the plot corresponding to the baseline ranking and the plot corresponding to the sort by download

remain steady in all three graphs. This suggests that a significant change in the quality of search results has major implication in how users interact with search results.

Additionally, the corresponding change in these 3 metrics is in accordance with the hypothesis, assuming that the ordering based on downloads is worse than an ordering based on baseline ranking.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this study we show how a change in the ordering of search results may result in change in user interaction. We were able to show that 3 of the metrics can measure significant change in user behavior with respect to change of ranking. By comparing a baseline model to two different types of ranking, we produce results that show the magnitude of change is proportional to the ordinal correlation between these rankings. With the degrade ranking being very close to the baseline ranking in terms of ordering, we see little change in the value of our metrics. However, when we change the ordering based on the number of downloads, the change in metric values are significant.

We can conclude the following in terms of rankings and user behavior:

1. Users are accustomed to finding their sounds in the first two pages of the search results page. In other words, their bias towards the top of the list leads them to be somewhat indifferent to look deep in the list. This is shown in the lack of change in two metrics that are related to the ranked position of the sounds.

2. A significant reordering of search results (from baseline to sort by download) had a direct impact on three of the metrics: Abandonment rate, queries per session and preview per query. On the other hand the other metrics did not monotonically change as a result of change in ranking.

3. Between the rank by score and rank by download, previews per query changes consistently but the preview before download remains the same.

4. Users of Freesound rely heavily on the SOLR's relevancy ranking feature. When the default ranking was changed to sort by download, more than half of the users reverted back to relevancy ranking as their choice of sorting the results.

5.2 Future Work

To further analyze the validity of click data as a source of evaluation of ranking quality, we consider performing more experiment with ordering of the results and computing the metrics. To confirm the hypothesis of this study, different ranking algorithms that change the ordering significantly must be undertaken. One of the challenges with Freesound is its position as a live heavily-used open platform. This makes changes to search functionality more difficult since some action might be intrusive to use experience.

We also plan to investigate other interaction exclusive to sound search as potential sources of implicit feedback. For example, the lenght at which a user preview a sound can be a strong indicator of how relevant a sound is to the query. Another direction to explore could be to combine previews and downloads as stronger or weaker measures of implicit feedback. If a user previews and downloads a sound in sequence that can be a stronger relevncy measure than only a single download.

Appendix A

Database Table Definitions

```
CREATE TABLE "clickthrough_query" (
    "id" serial NOT NULL PRIMARY KEY,
    "query_time" timestamp with time zone NOT NULL,
    "query_text" varchar(400),
        "sortby" varchar(100) NOT NULL,
        "advanced" varchar(1),
    "results_page_no" integer,
    "searchtime_session_key" varchar(32) NOT NULL,
    "results_shown" varchar(400) NOT NULL
);

CREATE TABLE "clickthrough_click" (
    "id" serial NOT NULL PRIMARY KEY,
    "click_datetime" timestamp with time zone NOT NULL,
    "click_type" varchar(2) NOT NULL,
    "authenticated_session_key" varchar(32),
    "searchtime_session_key" varchar(32),
    "sound_id" integer NOT NULL
);


ALTER TABLE clickthrough_click ADD COLUMN session_key varchar(32);
ALTER TABLE clickthrough_click ADD COLUMN query_id integer;
ALTER TABLE clickthrough_click ADD COLUMN session_id integer; -- to label
all the queries and clicks that occuring during a session
ALTER TABLE clickthrough_click ADD COLUMN rank_order integer;
ALTER TABLE clickthrough_query ADD COLUMN session_id integer;

/*
         post processing : cleaning some 'sort by' options that were parsed incorrectly
*/
update clickthrough_query SET sortby ='score desc' where sortby='%22score%20desc%22'
or sortby='%22score desc%22' or sortby='score%20desc';


/*
        Indexing
```

```
*/
CREATE INDEX ON clickthrough_click (authenticated_session_key);
CREATE INDEX ON clickthrough_click (searchtime_session_key);
CREATE INDEX ON clickthrough_click (click_datetime);
CREATE INDEX ON clickthrough_click (click_type);
CREATE INDEX ON clickthrough_query (searchtime_session_key);
CREATE INDEX ON clickthrough_query (query_time);
CREATE INDEX ON clickthrough_query (query_text);
CREATE INDEX ON clickthrough_query (sortby);
CREATE INDEX ON clickthrough_query (advanced);


/*
        Populating the session_key column in click table
*/
UPDATE clickthrough_click
SET session_key = searchtime_session_key
WHERE (authenticated_session_key = searchtime_session_key
AND authenticated_session_key != '' AND searchtime_session_key != '')
        OR (authenticated_session_key = '' AND searchtime_session_key != '');

UPDATE clickthrough_click
SET session_key = c2.searchtime_session_key
FROM clickthrough_click c2
WHERE  c2.authenticated_session_key != c2.searchtime_session_key AND
        c2.authenticated_session_key != '' AND c2.searchtime_session_key != ''
        AND (clickthrough_click.authenticated_session_key = c2.authenticated_session_key
        OR clickthrough_click.searchtime_session_key = c2.authenticated_session_key);

/*
        Populating the query_id column in click table
*/

UPDATE clickthrough_click
SET query_id = (select q.id
        from clickthrough_query q
        where (q.searchtime_session_key = clickthrough_click.authenticated_session_key OR
                q.searchtime_session_key = clickthrough_click.searchtime_session_key)
                AND q.query_time <= clickthrough_click.click_datetime
                AND q.query_time >= (clickthrough_click.click_datetime - interval '20 minutes')
        order by query_time DESC
        LIMIT 1);



/*
        Indexing
*/


CREATE INDEX ON clickthrough_click (session_key);
CREATE INDEX ON clickthrough_click (query_id);
```

```
/*
        Count the number of queries during each session. Each session is defined as a timespan
    whose activities are within 20 minutes of each other
*/
CREATE TABLE click_by_ts AS
        SELECT *
        FROM (
                SELECT text 'query' AS source, id, query_time AS click_datetime,
                        searchtime_session_key AS session_key
                FROM clickthrough_query
                UNION
                SELECT text 'click', id, click_datetime, session_key
                FROM clickthrough_click
        ) as cq
        ORDER BY cq.click_datetime;

CREATE INDEX ON click_by_ts (click_datetime);
CREATE INDEX ON click_by_ts (session_key);

--computing the gap between each click and filtering the ones more than 20 minutes apart.
CREATE TABLE session_terminating_clicks AS
    SELECT *
    FROM (
        SELECT a.source,a.id,a.click_datetime,a.session_key,(SELECT b.click_datetime
            FROM click_by_ts b
            WHERE b.session_key = a.session_key AND b.click_datetime > a.click_datetime
            ORDER BY b.click_datetime limit 1) - click_datetime as gap
        FROM click_by_ts a ) c
    WHERE c.gap > '00:20:00.0' ORDER BY c.click_datetime;

CREATE INDEX ON session_terminating_clicks (source);
CREATE INDEX ON session_terminating_clicks (id);
CREATE INDEX ON session_terminating_clicks (click_datetime);
CREATE INDEX ON session_terminating_clicks (session_key);
CREATE INDEX ON session_terminating_clicks (gap);


/*
        For every click_gap in session-terminating_clicks, in all the session_key matching
records that happen before the click_gap's click_datetime
*/
CREATE SEQUENCE seq1;
SELECT setval('seq1',1);

CREATE FUNCTION entersession() RETURNS void AS $$
DECLARE
        rec RECORD;
BEGIN
        FOR rec IN SELECT * FROM session_terminating_clicks LOOP
                RAISE NOTICE 'current value is ..%', (select currval('seq1'));
                UPDATE clickthrough_click c SET session_id=(select currval('seq1'))
                        WHERE c.session_key = rec.session_key
                                AND c.click_datetime <= rec.click_datetime
                                AND c.session_id is null;
                UPDATE clickthrough_query q SET session_id=(select currval('seq1'))
```

```
                        WHERE q.searchtime_session_key = rec.session_key
                                AND q.query_time <= rec.click_datetime
                                AND q.session_id is null;
                PERFORM nextval('seq1');
        END LOOP;
END;
$$ LANGUAGE plpgsql;

SELECT entersession();


create INDEX ON clickthrough_click (session_id);
create INDEX ON clickthrough_query (session_id);

with queries_first_100 as (select * from clickthrough_query order by query_time limit 1000)
select avg(session_id_count.ct) from (select session_id, count(*) as ct from queries_first_100 w

with queries_first_100 as (select * from clickthrough_query order by random() limit 1000) select

/*
        previews_per_query:
    Count the number of preview clicks for each query
*/
CREATE TABLE tq AS (SELECT *
        FROM clickthrough_query
        WHERE results_page_no = 1);
CREATE INDEX ON tq (searchtime_session_key);
CREATE INDEX ON tq (query_time);
CREATE INDEX ON tq (query_text);

CREATE TABLE associated_queries AS (
                SELECT q.id as a_queries,tq.id as ref_query, q.query_text
                FROM clickthrough_query q ,tq
                WHERE q.searchtime_session_key = tq.searchtime_session_key
                    AND q.query_time > tq.query_time
                    AND q.query_text = tq.query_text);
CREATE INDEX ON associated_queries (a_queries);
CREATE INDEX ON associated_queries (ref_query);

select avg(count)
from (
        SELECT associated_queries.ref_query, count(*)
        FROM  clickthrough_click c, associated_queries
        WHERE c.click_type = 'sp'
                AND c.query_id=associated_queries.a_queries OR
c.query_id=associated_queries.ref_query
        GROUP BY associated_queries.ref_query
        ORDER BY count DESC) as query_prev_count;

/*
        previews_per_download_per_SERP:
        For each download, count the number of previews before that download and the time
the SERP containing the downloaded sound was requested (pagination included)
*/
```

```
select c1.id, count(*)
from clickthrough_click c1, clickthrough_click c2
where c1.session_key = c2.session_key
    and c1.query_id = c2.query_id
    and c1.click_type = 'sd'
    and c2.click_type='sp'
    and c2.click_datetime < c1.click_datetime
    and c2.click_datetime > (select q.query_time from clickthrough_query q where c1.query_id = q
group by c1.id;


/*
        max_reciprocal_rank
        For each query, find the rank of the highest ranked download
*/
create table q1 as (select  *
            from clickthrough_query
            where results_page_no=1);
create index on q1 (searchtime_session_key);
create index on q1 (query_time);
create index on q1 (query_text);

create table qs as (select q1.id as qid,q2.id as allqueries
        from clickthrough_query q2,q1
        where q2.searchtime_session_key = q1.searchtime_session_key
            and q2.query_text = q1.query_text
            and q2.query_time >= q1.query_time
            and q2.query_time < q1.query_time + interval '30 minutes'
        order by q1.id,q2.query_time);
create index on qs (qid);
create index on qs (allqueries);

select qs.qid,min(c.rank_order)
from clickthrough_click c,qs
where c.query_id = qs.allqueries
group by qs.qid;

--------------------------------------------------------------------------------



CREATE FUNCTION rank_extractor (results text, sound_id integer, page_no integer)
  RETURNS integer
AS $$
  import ast
  try:
    return ((page_no -1)*15)+(ast.literal_eval(results).index(int(sound_id))+1)
  except:
    return 0
$$ LANGUAGE plpythonu;


update clickthrough_click set rank_order = (select rank_extractor
        (q.results_shown, sound_id, q.results_page_no))
        from clickthrough_query q where query_id = q.id;
```

```
update clickthrough_click set rank_order = null where rank_order = 0;

with query_preview_counts as
        (select query_id, count(*) as count_of_previews
        from clickthrough_click
        where click_type='sp' group by query_id),
        query_download_counts as (select query_id, count(*) as count_of_downloads
                        from clickthrough_click
                        where click_type='sd' group by query_id),
        query_preview_and_download_counts as
                (select t1.query_id,t1.count_of_previews,t2.count_of_downloads,
1.0*t2.count_of_downloads/t1.count_of_previews as ratio
                from query_preview_counts as t1, query_download_counts as t2
                where t1.query_id = t2.query_id)
select avg(ratio)
from query_preview_and_download_counts;
```

Appendix B

Queries for computing The Metric

B.1 Abandoment Rate

```
with query_pairs as
(select *
from ( select q1.searchtime_session_key, q1.query_time,
(SELECT query_time FROM clickthrough_query
where clickthrough_query.searchtime_session_key = q1.searchtime_session_key
and query_time > q1.query_time
order by clickthrough_query.query_time limit 1) AS next_query_time
from (select * from clickthrough_query
where sortby='score desc' and advanced='0'
order by random() limit %s) as q1 ) as qps
where qps.next_query_time is not null)
select count(*)
from ( select qp.searchtime_session_key,
(select count(*) from clickthrough_click c
where c.session_key = qp.searchtime_session_key
and c.click_datetime > qp.query_time
and c.click_datetime < qp.next_query_time) as count_of_dnd_snds
from query_pairs qp ) as qc where qc.count_of_dnd_snds = 0
```

B.2 Query Per Session

```
select session_id, count(*) from clickthrough_query where sortby='score desc' and advanced='0' a
```

B.3 Previews Per Query

```
select qs.id,qs.first_query_time, qs.next_query_time, count(*) from clickthrough_click c, (selec
```

B.4 Previews Before Download

```
SELECT qs.id,qs.first_query_time, qs.next_query_time, COUNT(*)
FROM clickthrough_click c,clickthrough_click c2,
(SELECT q1.id, q1.searchtime_session_key, q1.query_time AS first_query_time,
( SELECT q2.query_time FROM clickthrough_query q2
WHERE  q2.sortby = 'num_downloads desc' AND q2.results_page_no = 1 AND q1.searchtime_session_key
FROM clickthrough_query q1
WHERE q1.results_page_no = 1
AND q1.sortby = 'score desc' order by random() limit %s)
 AS qs WHERE c.click_type = 'sp' AND c2.click_type = 'sp' AND
c.click_datetime > qs.first_query_time AND c.click_datetime < qs.next_query_time
AND (c.searchtime_session_key = qs.searchtime_session_key
OR c.authenticated_session_key = qs.searchtime_session_key)
AND (c2.searchtime_session_key = qs.searchtime_session_key
OR c2.authenticated_session_key = qs.searchtime_session_key)
AND c2.click_datetime > qs.first_query_time
AND c2.click_datetime < qs.next_query_time
GROUP BY qs.id,qs.first_query_time, qs.next_query_time
```

B.5 Maximum Reciprocal Rank

```
select * from (select q.id, 1.0/min(c.rank_order) as max_recip_rank from clickthrough_click c ,
```

B.6 Mean Reciprocal Rank

```
select * from (
select q.id, sum(1.0/c.rank_order) as summation from clickthrough_click c, (select id from click
```

Bibliography

[1] Chidansh Amitkumar Bhatt and Mohan S Kankanhalli. Multimedia data mining: state of the art and challenges. *Multimedia Tools and Applications*, 51(1):35–76, 2011.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley, second edition, 2011. ISBN 9780321416919. URL http://books.google.es/books?id=HbyAAAAACAAJ.

[3] Ricardo Baeza-Yates, Andrei Z Broder, and Yoelle Maarek. The new frontier of web search technology: seven challenges. In *Search computing*, pages 3–9. Springer, 2011.

[4] Ricardo Baeza-Yates, Carlos Hurtado, Marcelo Mendoza, and Georges Dupret. Modeling user search behavior. In *Web Congress, 2005. LA-WEB 2005. Third Latin American*, pages 10–pp. IEEE, 2005.

[5] Bernard J. Jansen and Amanda Spink. An analysis of web searching by european alltheweb.com users. *Information Processing and Management*, 41(2):361 – 381, 2005. ISSN 0306-4573. doi: 10.1016/S0306-4573(03)00067-0. URL http://www.sciencedirect.com/science/article/pii/S0306457303000670.

[6] Daniel E Rose and Danny Levinson. Understanding user goals in web search. In *Proceedings of the 13th international conference on World Wide Web*, pages 13–19. ACM, 2004.

[7] Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *Proceedings of the 14th international conference on World Wide Web*, pages 391–400. ACM, 2005.

[8] Yiqun Liu, Min Zhang, Liyun Ru, and Shaoping Ma. Automatic query type identification based on click through information. In *Information Retrieval Technology*, pages 593–600. Springer, 2006.

[9] Ricardo Baeza-Yates, Liliana Calderón-Benavides, and Cristina González-Caro. The intention behind web queries. In *String processing and information retrieval*, pages 98–109. Springer, 2006.

[10] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Yong Yu, Wei-Ying Ma, WenSi Xi, and WeiGuo Fan. Optimizing web search using web click-through data. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 118–126. ACM, 2004.

[11] Neela Sawant, Jia Li, and James Z Wang. Automatic image semantic interpretation using social action and tagging data. *Multimedia Tools and Applications*, 51(1):213–246, 2011.

[12] James Surowiecki. *The wisdom of crowds*. Anchor, 2005. ISBN 978-0-385-50386-0.

[13] Helen Ashman, Michael Antunovic, Christoph Donner, Rebecca Frith, Eric Rebelos, Jan-Felix Schmakeit, Gavin Smith, and Mark Truran. Are clickthroughs useful for image labelling? In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 191–197. IEEE Computer Society, 2009.

[14] Klimis Ntalianis, Anastasios Doulamis, and Nicolas Tsapatsoulis. Implicit visual concept modeling in image/video annotation. In *Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, pages 33–38. ACM, 2010.

[15] Donn Morrison, Theodora Tsikrika, Vera Hollink, Arjen P de Vries, Éric Bruno, and Stéphane Marchand-Maillet. Topic modelling of clickthrough data in image search. *Multimedia Tools and Applications*, pages 1–23, 2012.

[16] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117, 1998.

[17] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[18] Ricardo Baeza-Yates and Yoelle Maarek. Usage data in web search: benefits and limitations. In *Scientific and Statistical Database Management*, pages 495–506. Springer, 2012.

[19] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[20] Joseph John Rocchio. Relevance feedback in information retrieval. 1971.

[21] Amanda Spink, Bernard J Jansen, and H Cenk Ozmultu. Use of query reformulation and relevance feedback by excite users. *Internet research*, 10(4):317–328, 2000.

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

[23] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10. ACM, 2006.

[24] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.

[25] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161. ACM, 2005.

[26] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. In *ACM SIGIR Forum*, volume 37, pages 18–28. ACM, 2003.

[27] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 87–94. ACM, 2008.

[28] Gavin Smith, Chris Brien, and Helen Ashman. Evaluating implicit judgments from image search clickthrough data. *Journal of the American Society for Information Science and Technology*, 63(12):2451–2462, 2012.

[29] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169. ACM, 1993.

[30] Jinxi Xu and W Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11. ACM, 1996.

[31] Hinrich Schütze. Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123, 1998.

[32] Kamal Ali and C Chang. On the relationship between click-rate and relevance for search engines. *Proc. of Data-Mining and Information Engineering*, pages 213–222, 2006.

[33] Rodrigo B. Almeida and Virgilio A. F. Almeida. A community-aware search engine. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 413–421, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X. doi: 10.1145/988672.988728. URL http://doi.acm.org/10.1145/988672.988728.

[34] Filip Radlinski, Madhu Kurup, and Thorsten Joachims. How does clickthrough data reflect retrieval quality? In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 43–52. ACM, 2008.

[35] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.