Birger Topphol

Proportionality for Diversifying Information Retrieval on Structured Data

June 2020

Master's thesis

Master's thesis

2020

Birger Topphol

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Norwegian University of
Science and Technology

Proportionality for Diversifying Information Retrieval on Structured Data

Birger Topphol

Master of Science in Informatics
Submission date: June 2020
Supervisor: Trond Aalberg

Norwegian University of Science and Technology
Department of Computer Science

Abstract

Diversity helps resolve several problems in information retrieval (IR) and information filtering (IF), and has been shown to increase user satisfaction. For IR the main problems are ambiguous queries and duplicative data, while IF is most concerned with supplying the user with a variety of choices.

The past few years there has been increasing interest in using diversity and other beyond-accuracy objectives in IR and IF. This increased interest has resulted in a variety of approaches to improving diversity in search results and recommendations. However there is still a lack of diversity focused research for IR, on structured and semi-structured data.

This thesis contributes to filling that hole, by investigating what methods can be used to increase diversity when searching structured data.

A query combination approach is presented, using the Sainte-Laguë method to select high ranking items from a set of results obtained by querying different fields of the data set. A case based study performed on a music database, serves as a proof of concept, demonstrating the behaviour of the method, and showing that it is a viable option for mitigating the effects of ambiguous queries, and duplicative data.

Samandrag

Mangfald hjelp til med å løyse fleire problem innan informasjonsgjennfinning og informasjonsfiltrering, og har vist seg å gjere brukarar meir fornøgde. Hovudproblema for gjennfinning er tvetydige spørjingar og duplikative data, mens filtrering er mest oppteken av å tilby brukeren eit variert utvalg.

Dei siste åra har det vore aukande interesse for å bruke mangfald og andre mål utover nøyaktigheit i gjennfinning og filtrering. Denne auka interessa har resultert i ei rekkje tilnærminger for å forbetre mangfaldet i søkjeresultat og anbefalinger. Det er imidlertid framleis mangel på forskning med fokus på mangfald for gjennfinning på strukturerte og semistrukturerte data.

Denne oppgava bidreg til å fylle det holet, ved å undersøkje kva metoder som kan nyttast til å auke mangfaldet ved søk i strukturerte data.

Ei tilnærming vert presentert, som nyttar spørringskombinasjon ved bruk av Sainte-Laguëmetoden for å velgje høgt rangerte element frå eit sett resultat oppnådd ved å søkje i ulike felt i datasettet. Ei casebasert studie utført på ein musikkdatabase fungerar som "proof of concept", demonstrerar oppførselen til metoda, og viser at det er eit aktuelt alternativ for å dempe effekten av tvetydige spørjingar og duplikative data.

Contents

1 Introduction **1**

2 Theory **3**

2.1 Information Retrieval . . . 3

2.1.1 Filtering vs Retrieval . . . 4

2.2 Beyond-Accuracy Objectives . . . 5

2.2.1 Coverage . . . 5

2.2.2 Serendipity . . . 6

2.2.3 Novelty . . . 6

2.2.4 Diversity . . . 7

2.3 Information systems . . . 8

2.4 Diversity in information systems . . . 8

2.4.1 Diversity based recommenders . . . 9

2.4.2 Diversity based search . . . 9

2.4.3 Different domains . . . 10

2.4.4 Proportionality . . . 10

2.4.5 Data fusion and query combination . . . 11

2.4.6 Structured data . . . 11

2.5 Evaluation . . . 12

2.5.1 Implicit vs explicit feedback . . . 12

2.5.2 Offline accuracy metrics . . . 12

2.5.3 Offline diversity metrics . . . 13

2.5.4 Online evaluation . . . 15

3 Methodology **16**

3.1 Proposed method . . . 16

3.2 Test data . . . 17

3.3 Implementation . . . 18

3.4 Experiment . . . 20

4 Case study **22**

4.1 Basic Sainte-Laguë . . . 23

4.2 Sainte-Laguë with multipliers 23

4.3 Sainte-Laguë with score caps 26

4.4 Combining caps and multipliers 28

5 Summary **32**

6 Future work **34**

Bibliography **36**

A Final ranking results **41**

1 Introduction

Information retrieval (IR) [2] has become a vital part of our society. We use online search engines on a daily basis through work and in our spare time. It is so readily available through our smartphones that most people do not give a second thought to what goes into the process of satisfying their information need.

To a lot of people searching is synonymous with web search or Googling. Although it is true that, this is the most widely used form of search, there are many other types of search that are used in a variety of domains. Some focus on recall, by providing every potentially relevant result for a search query, while others focus on precision, by ensuring that every result that is retrieved is as relevant as possible, but most methods have a trade-off between the two. Recall and precision form the basis for accuracy metrics, which for a long time has been the primary way researchers have evaluated their IR methods. In the last decade or so, this has started to shift towards focusing on measurements that go beyond the traditional accuracy metrics, including diversity, novelty, serendipity and coverage. This shift may in large part be motivated by the rise of multimedia streaming services and the recommender systems that go along with them. Movie streaming services like Netflix have invested a significant amount of money in order to provide satisfactory recommendations for its users. Recommender systems are one of the main technologies of information filtering (IF), which is very much related to IR, and often considered to be a part of it.

Even with significant amounts of research related to these beyond-accuracy objectives, there is still a lack of commonly agreed upon metrics or even definitions for them [11]. Despite the disagreements on the best ways to measure diversity, there is a consensus on the importance of diversity in both IR and IF. When it comes to recommender systems, it is quite obvious that a collection with a bit of variety would be better than a collection of near identical items. The two main reasons for diversifying search results are resolving ambiguous queries and avoiding duplicative information. If a user does not know exactly what to search for, or the query term has multiple meanings, the goal of a diverse search result is to ensure that the user still finds at least one relevant item [1].

The vast majority of the research on diversity has been on pure text documents or pure text representations of web pages. There has been shown little interest in diversifying structured or semi-structured data. Collections like the TREC[1] data sets are one major contributor to the increased interest in diversification, giving researchers common goals to work towards. Perhaps the lack of interest for structured data is in part due to the lack of such well known and commonly used structured data sets, with accompanying diversity goals.

This thesis aims to address the lack of research on diversity in structured data, by presenting a query combination method that increase diversity by promoting proportionality of results from querying different fields of the data. The research questions that are answered in this thesis are the following:

RQ1: What method can be used to improve diversity in IR on structured data?

RQ2: How does the chosen method behave?

The thesis starts of with a brief presentation of information retrieval and the closely related field information filtering. It then goes on to give an overview of beyond-accuracy objectives, with the main focus on diversity. Some of the methods for increasing diversity in recommender systems and search engines are presented, followed by methods and evaluation metrics used to measure diversity in information retrieval and filtering. After this, the new method is presented and a case based evaluation serves as a proof of concept in order to highlight some of the systems behaviour. Finally there is a brief discussion of strengths and weaknesses of such an approach and possible future directions to continue the research.

[1] `https://trec.nist.gov/`

2 Theory

2.1 Information Retrieval

Information retrieval (IR) [2] is a broad field of computer science where the primary goal is to find items in a collection that satisfy a users information need. IR may be performed on text documents, web-sites, metadata or media files like images, video and audio. Most search engines today are performed on full-text indexes. The intention of using such indexes is to more quickly determine whether an item is relevant to a given query. Often they contain information on the frequency of words in the documents of a collection.

The user providing a query to the IR system is a part of, and what starts the IR process. It is therefor important for IR researchers to understand users behaviour and approach to satisfying an information need, as well as understanding what makes a user satisfied with a search engine. Most users start off by providing a fairly general query and go on to be more specific if the search engine does not provide the desired information. Studies have shown that users are more satisfied if the results include a greater variety of items, even tho they might be less relevant to the query overall [35, 30, 9].

One of the research fields within IR on structured data is Entity-oriented search [3]. The idea behind it is to retrieving entities instead of documents when performing search. These entities can then be listed as the final result, or be used to retrieve documents that are linked to these entities. Entities are basically individual things, like a person, a movie or an animal. These entities have attributes and may have links to other entities. Wikipedia[2] is a good example of a collection of entities. Retrieval on collections can be done through constructing entity representations by gathering information from available documents or predefined knowledge bases, and then perform traditional document based IR. For Wikipedia this could involve simply flattening all the fields of the entities page, but could be significantly more complex for other collections. Another approach is to create methods that are specialized for retrieval directly on (semi-)structured data.

[2] https://www.wikipedia.org/

2.1.1 Filtering vs Retrieval

Information filtering (IF) is often used in conjunction with information retrieval, but they are not quite the same. These terms are very closely related as discussed in detail by Belkin and Croft in 1992 [5], and by Hanani et al. in 2001 [15]. The main goal for both retrieval and filtering is essentially the same, which is to fulfil a users information need. The difference lies in the approach to achieve this goal, and the context in which they operate. IR is most often thought of as the process of finding a set of relevant data in a much larger set of data. IF on the other hand can be thought of as removing data that is not relevant to the user. IR methods are applied whenever the user has a short-term information need, while IF is more of a passive process that works in the background to satisfy a long-term information need.

The most prevalent form of IR is undoubtedly web search. IF is related to a series of technologies, like filtering emails and signal processing, but the one that most closely resembles information retrieval is recommender systems. This is especially true when it comes to evaluating based on beyond-accuracy objectives, like diversity and novelty, discussed later.

IR and IF have both started adopting techniques from the other field, and in a sense are starting to merge together. Hanani et al. [15] list a set of differences between the fields. Some of these still apply, but others are not as relevant any more. For instance the database usage. They claim that IR works on static databases, while IF handle constantly changing data or streams of data. Recommender systems for movies and books work on relatively static databases. It is also not very common to provide a recommender system, while not providing the ability to search in the same data.

The idea that the users are unknown to an IR system, is also not always the case any more. One of the main techniques used in IF is building a profile based on previous activity and filtering out entries that do not match well with this profile. Traditionally this technique was not used in IR, but has been adopted by search giants such as Google, although not without controversy. For recommender systems one can just as easily argue that the process is finding entities that match well with the profile, matching the definition of IR.

2.2 Beyond-Accuracy Objectives

Traditionally search algorithms in IR, as well as recommender systems, were measured by the relevance of each individual item in the result set. While relevance is still of vital importance, it is no longer the only consideration when constructing new methods [23]. The relevance of each result in a ranking can in most cases not be assumed to be independent of other documents in the ranking [31]. If search is performed on a large collection of documents with a high degree of duplicate information, assuming independent relevance may lead to all the results giving essentially the same information. There might be multiple meanings to some queries, like for instance "rock", which could mean the musical genre, hard objects or many other meanings. In this case it could be greatly beneficial to provide some results for each of the different meanings.

Performing search on a set with many similar items and considering only relevance, will most likely give a result set where each item is very similar to every other item. Diversifying the list may help in this situation, but too much diversity may also cause problems. If a list contains only items that have nothing to do with each other, and are not similar in any way, it is highly unlikely that all of them are relevant to the user, or the query they supplied.

These problems are not possible to address purely with relevance. Therefor many authors have started investigating methods and evaluation metrics that go beyond maximizing accuracy. The aptly named beyond-accuracy objectives include diversity, novelty, serendipity, coverage, and many others. All of these objectives are defined in slightly different ways depending on which article you read, and are often overlapping. A brief description of these four objectives will be presented below. A more thorough survey relating to recommender systems was conducted by Kaminskas and Bridge [18].

2.2.1 Coverage

In some circumstances the ability to supply the users with as wide a range of items as possible, is very important. Consider an online book store that only advertises the same 10 books all the time and to every user. This would most likely lead to all other books getting a lot less sales than these 10. From a business perspective, this would probably not be a good idea. A better idea would be to design the system to cover as much of their catalogue as possible.

Coverage for recommender systems it is most often defined as the ratio of the collection that is possible for the system to recommend [35], or as the ratio of the collection that are actually returned as recommendations [18].

Yates and Neto [2] uses a definition of coverage in IR that is the ratio of items the user knows about and are retrieved by the system. By this definition, increasing coverage would mean reducing novelty, and may also decrease diversity (these terms are discussed later).

Coverage can also be defined in relation to clusters or subtopics of items rather than of individual items [33]. When increasing coverage, by this definition, there wouldn't necessarily be a decrease in novelty and diversity. In fact it would more likely promote diversity by covering more subtopics.

2.2.2 Serendipity

Serendipity refers to the surprise of an item being included in the result set or recommendation, while still being relevant [16]. This is the ultimate goal of any recommender system, and may in many cases also be beneficial for search engines. If a set of recommendations only contains items that the user knows about, the recommendations are not very useful, and the system looses its purpose. On the other hand if the recommender provides only very obscure objects that the user has never heard about, but doesn't like either, the system has also failed. Finding a balance between these two is the goal of serendipity.

Precisely measuring surprise without getting explicit feedback from users is very difficult, if at all possible. Even with the explicit feedback it might not be so easy to determine whether the system reaches its goal or not. Users can misinterpret the objective of this feedback or use it in vastly different ways. Ultimately, serendipity is also reliant on the users expectations and knowledge of the domain. An item that is surprising to some users may be obvious to others.

2.2.3 Novelty

A very closely related, but more measurable objective is novelty. That is the measure of the rarity of an item, or how much an item is different from what has been seen before, either by a single user or the entire system. Most authors claim that serendipity and novelty are too similar to each other and therefor only consider novelty, or never mention serendipity at all.

A common way to think of increasing novelty is to promote the long-tail items. Vargas and Castells [28] differentiates between popularity-based and distance-based novelty. Popularity-based novelty is defined as the inverse of the likelihood of an item being previously observed in a set of recommendations. This means that promoting items that are less likely to have been seen, and therefor are in the long-tail, will increase novelty of the recommendation. Distance-based novelty is defined as the minimum distance between a given item and the items already in the recommendation set. The distance measure can be calculated from for instance a content-based or collaborative-filtering similarity measure. The latter making it intuitively similar to popularity-based novelty.

2.2.4 Diversity

Diversity is often defined based on the average pairwise dissimilarity of items in the search result or recommendation set [35, 28, 18]. This is a fairly straight forward definition where diversity is high when every item is different from all other items. The difficult part is determining what makes items similar.

Vee et al. [29] defines a diversity ordering for structured data, where items are more diverse if they are different on attributes that have higher priority. When searching in a collection of cars, the make and model of the car may be considered more important than color or year. Their main idea is that if a user searches for Honda, the result should include as many models of Honda as possible. If the result includes multiple of one model, they should have different color, and so on. This results in a diversity definition where cars whose only difference is their model is considered more diverse than if the only difference is their color. Such an ordering can be defined by a domain expert or even be specified by the users.

Diversity is often used where it could be more accurate to call it novelty, serendipity or any other beyond accuracy-objective. Most definitions overlap either partially or completely with the other objectives. Clarke et al. [11] however describes a distinction between them: novelty – the need to avoid redundancy - and diversity – the need to resolve ambiguity. For this thesis the goal of diversity will be defined as both the need to avoid redundancy and the need to resolve ambiguity, to avoid having to mention both diversity and novelty.

2.3 Information systems

One of the most widespread methods for determining the importance of a word in a document, is *TF-IDF*, which stands for term frequency–inverse document frequency. Term frequency refers to the number of times a term appears in a document. Document frequency is the number of documents in a collection that contains the given term. This means that words that appear a few times in one document, but rarely in others is considered more important than one that appear often in the same document, but also is used in many other documents. When entering the query "rock and roll" into a search engine that uses TF-IDF, the term 'and' will probably not have very much impact on what documents are included in the result.

Another ranking function in widespread use, is *BM25* [25], which was first introduced by Okapi at TREC in the 90s, but has evolved a lot since. BM25 is based on the traditional probability ranking principle [24], but extends it to take term frequency and document length into consideration. Since its introduction there has been many variations. Robertson et al. [26] proposed an extension, later known as *BM25F*, that includes weighted fields for use on structured or semi-structured data.

2.4 Diversity in information systems

Diversification in information retrieval and information filtering have many similarities. The basic principle is the same; relevance of a set of items is not only dependent on the relevance of the individual items, but also on their relations to each other [8, 31]. This means that filling a set of recommendations or a list of search results with only the most individually relevant items is not always the best approach.

The reasoning behind this need for diversity are slightly different. A recommender system is there to satisfy a users long term information needs. For instance the need for good movies to watch or the latest news relating to their interests. Recommender systems need to have a variety of items for the user to choose from [23]. A recommender that only recommends songs by the same artist might be highly accurate, but is not very useful. For search engines ambiguous queries and duplicative data are the most pressing issues to address using diversity.

2.4.1 Diversity based recommenders

The past decade has seen a huge increase in popularity in online video streaming services, and with it an increasing focus on improving recommender systems. A lot of the new research relate to beyond-accuracy objectives. More specifically to increase diversity and promote more novel recommendations.

Ziegler et al. [35] present a method for diversifying a set of recommendations based on a classification taxonomy of the data. Their algorithm creates a second ranking based on the dissimilarity of the candidate items to the items already in the set. This ranking is then merged with the original, relevance based ranking, to produce the next recommended item. The process is repeated until the desired number of recommendations is reached. Their results show that this method decreases the average relevance of the individual items, but users preferred the lists with more diversity.

There are large individual differences in users preferences when it comes to novelty and diversity in recommender systems. Kapoor et al. [19] recognizes this and present a method for dynamically tuning the system according to the behavior of the user.

2.4.2 Diversity based search

Diversity in search may not have received the same enthusiasm as recommender systems have, in the past few years, but there has not been a complete lack of interest. The importance of diversification in search was already recognized by Goffman in 1964 [14]. While the most important aspect of search still remains relevance to the query, methods that rely solely on relevance will have problems that can be addressed with diversity. Ambiguous queries are almost always a concern and some domains can benefit from having some novel and serendipitous results. When searching for 'Bob Dylan' in a music database it is highly unlikely that the preferred result is a list of all his compilation albums simply named 'Bob Dylan'.

Chen and Karger [10] study the effects of diversification by applying a greedy ranking method where they sequentially select items that are relevant, but different from the ones already in the result set. The intuition behind their approach is that all the items that has already been selected are no longer relevant, so the next chosen item should not be similar to any of those. This approach increases the likelihood of finding at least one relevant item among the top ranking results, compared to a pure relevance ranking.

Zhai et al. [31] approach diversification in information retrieval by assigning subtopics to each document, and making sure that the result set includes as many of these subtopics as possible.

Zheng et al. [33] also considers diversity in relation to subtopics. They approach diversity as a problem of increasing coverage, and apply coverage functions in a greedy algorithm.

2.4.3 Different domains

Search engines and recommender systems are not domain independent. In fact different domains may have vastly different requirements for systems that work on their data [18]. Take for instance a search engine used by a court in criminal cases. Focusing on increased diversity by trading off relevance can have grave consequences for the convicted part. Even within fairly similar domains where diversity and novelty is desired, it may not apply to all domains to the same degree [19]. For a music streaming service it may be fine, and even preferred, to include some songs that the user knows about, while recommending news articles that the user has already read makes little sense.

In resent years there have been increasing research done on diversifying very specific domains and use cases. Zhang and Setty [32] study diversification through retrieval of comments on social media. Cai et al. [7] investigate diversity in query auto-completion and propose a method for reducing duplicity in suggested queries.

2.4.4 Proportionality

Dang and Croft [12] take a different perspective on diversity in search by promoting proportionality. Their approach (PM-2) uses the Sainte-Laguë method in order to select a proportional representation of items, with regards to a set of predefined subtopics. Their results indicate that promoting proportionality naturally increases diversity by minimizing redundancy.

In parliamentary elections Sainte-Laguë is a method for allocating seats to competing political parties proportionally to their vote count [13]. There are a range of other methods in use by different parliaments, with varying degree of favorability towards small or large parties. The Sainte-Laguë slightly favors small parties.

2.4.5 Data fusion and query combination

Data fusion and the closely related query combination is the process of taking multiple ranked lists and merging them together to create a final ranked list. For query combination the ranked lists are produced by multiple interpretations of the query using the same ranking method. Data fusion on the other hand uses the same query on different ranking methods.

There are several ways to decide the order of the new ranking. Two of the simplest are purely going by the rank of the original lists, or purely by the items score in the original lists. More advanced methods may apply a combination of the two, or use other techniques to modify the initial scores or ranks.

Belkin et al. and Fox & Shaw [4] report on two separate studies on query combination, that show improvements on relevance based metrics, compared to any of the single-query rankings. They also fuse their two systems and find that this further improves the performance, indicating that data fusion has similar benefits as query combination.

Liang et al. [21] investigate what effects data fusion has on diversity, and find that it often improves diversity, even when the fusion methods are designed without diversity in mind. They also present their own diversity data fusion (DFF) method, based on PM-2 [12], that significantly outperforms other data fusion methods on a range of diversity metrics.

2.4.6 Structured data

There has been very little research done in relation to diversity in information retrieval on structured data. One of the main reasons for this may be the lack of well defined data sets that researchers can use to compare their findings to other research. Another reason may be the inherent difficulty of defining a satisfactory evaluation metric for diversity in structured data. Structured data is highly dependent on the domain, meaning that it inherits all the problems related to diversity in different domains.

One approach that does use structured data is proposed by Vee et al. [29], where items are represented by a set of features. The features are ordered by priority. The main idea is to go through the set of items and swap out items in the result set by items that differ on high priority features. Their main focus is on efficiency of finding the optimal diverse result set, rather than a trade-off between diversity and relevance.

2.5 Evaluation

Evaluation of search engines and recommender systems can generally be divided into two groups; offline and online evaluation. Offline evaluations are objective functions based on the content of the data, such as measuring the difference between texts, word for word. Online evaluation is based on explicit or implicit feedback from users of the system. It is widely recognised that online evaluations are more precise than offline evaluations, when it comes to judging user satisfaction [18], but the amount of time and effort needed to perform them makes offline evaluation still an important tool.

2.5.1 Implicit vs explicit feedback

Offline evaluation on recommender systems are most often conducted on previously collected data on users preference, through explicit or implicit feedback. Explicit feedback means that users actively express their satisfaction, by rating a movie or liking a news article. Implicit feedback is typically when a user clicks on a given recommendation or search result. This means that explicit feedback will be more sparse, while implicit feedback will have more noise. It is also more difficult to determine if a user is satisfied with a recommendation when the user merely clicked on it.

Collecting explicit feedback from search engines is not a very common practice, and with good reason. When searching for some information on the internet, the user will usually not know if a given search result will provide this information until they enter the site from the result, and thus leave the search engine. Most users would not bother going back to the result page in order to give feedback on the relevance of the result. Thus offline evaluation on search has to rely on implicit feedback, if it is based on user behaviour at all. A user clicking on a result, and then not clicking on another result, may be interpreted as the users information need being satisfied.

2.5.2 Offline accuracy metrics

Some of the most widespread metrics for offline evaluation of both recommender systems and search engines are based on the assumption that the relevance of items are independent of other items in the collection. Herlocker et al. [16] has provided an extensive survey on accuracy metrics for evaluation of recommender systems, most of which may also be used for search engines.

Two basic building blocks for many evaluation metrics are precision and recall. Precision is the fraction of retrieved items that are relevant. Recall is the fraction of relevant items that are retrieved.

There have been several proposed metrics to combine precision and recall into a single value measure. *Mean average precision (MAP)* is one of the most widely used. The idea is to calculate the average of precision values obtained for each relevant document in the ranking.

A problem with recall and precision is that it does not allow for graded relevance of items. This means that a result set filled with partially relevant items may be considered just as good as a set of highly relevant items. Järvelin and Kekäläinen [17] recognize this problem and propose the *Normalized Discounted Cumulative Gain (nDCG)*. For a given query, a set of information nuggets are defined. The basic idea of nDCG is to reward including items that are relevant to several of these information nuggets, while discounting items for being ranked lower.

As collections grow larger it is increasingly unlikely to have relevance judgement for all items. Buckley and Voorhees [6] show that widespread evaluation metrics are not robust on incomplete relevance judgement, and introduce a metric to solve this problem, called *bpref*. Instead of using the absolute rank of relevant documents, bpref uses the number of judged non-relevant items that rank before relevant items. This means that including more items that have not been judged does not affect the bpref score, but it may reduce the score of other metrics.

2.5.3 Offline diversity metrics

There is no consensus on what makes a good set of recommendations or a good ranking of search results when diversity and novelty is considered [11]. This, of course, makes it impossible to agree upon a general purpose evaluation metric for diversity and novelty. There has still been several attempts at creating evaluation metrics that go beyond the traditional relevance-based metrics. Some try to define more general purpose evaluation metrics, while others are focusing more on tailoring it to their specific domain and use case.

In addition to the survey on accuracy metrics, Herlocker et al. [16], also surveyed a range of beyond-accuracy metrics, but a lot has changed since when the study was published. Maksai et al. [22] provide a more up to date investigation on the topic.

One of the earliest proposed metrics that went beyond accuracy is the *Maximal Marginal Relevance (MMR)* [8]. The main principle is to maximize relevant novelty, which means that they maximize relevance while keeping redundancy at a minimum. Each new element that is considered in the ranking, will be penalized for being similar to items that have already been ranked. A lot of later research have used this as starting point or motivation for more elaborate metrics.

Intra-list similarity metrics calculate the aggregate similarity of items within a set of recommendations or top-N search results, using some similarity metric. A version of it was presented by Ziegler et al. [35]. A high intra-list score means that items have high similarity and hence low diversity.

Lathia et al. [20] propose a metric that depends on how many new items are presented during different visits. The metric compares a set of recommendations to the set of all previous recommended items for a given user. A high temporal diversity may be desired for news recommenders and other rapidly changing domains.

Often a combination of accuracy and diversity metrics provide results that more closely resemble user behaviour [22]. A trade-off between them can be used to tune the system for different preferences [34].

The above metrics do not consider the rank of items, and may therefor need modifications to be applicable to IR. Vargas and Castells [28] argue that also evaluations on recommender systems should take rank into account, because users are more likely to consider items towards the top. They propose a framework that unifies and generalizes several metrics and propose new metrics.

Clarke et al. [11] consider diversity in an IR setting, and propose a diversity-aware evaluation metric. They build their measure on nDCG, and extend it to penalize items that provide the same information nuggets as items that rank higher.

None of these metrics have gained enough traction to be universally accepted as the gold standard for diversity evaluation. Even less can they be considered good estimation of user satisfaction, as exemplified in a study by Said et al. [27]. There seem to be a trend of tailoring recommenders and search engines to their specific domains and user patterns, and so most authors define their own way of evaluating the usefulness of their system.

2.5.4 Online evaluation

Given this gap between user satisfaction and performance when it comes to offline metrics [27], the natural course would be to perform online evaluation, by letting users test the system and give their opinions, or analyze their behaviour.

The main drawback of performing online evaluation is the amount of time and resources required to go through with it. For an experiment to have real significance it needs to be performed on a large group of people. This can be very costly and time consuming.

The most useful experiments are when users are unaware that they are participating in them [18]. When users do not know about the experiment, they will have a more authentic behavior, making their feedback more accurate.

When performing a more qualitative study, by performing case based evaluation, it is important to draw knowledge from past results. One of the main things to take away from studies on diversity and real users, is that light diversification seem to give the most satisfaction. Multiple studies on recommendation systems show that users prefer some diversification, but when the recommendations reach a certain level of diversity, the benefit of more diversity diminishes [35, 30, 9].

Castagnos et al. [9] also show that additional information on why an item is recommended may sometimes be needed. Without the extra information the item might be perceived as not relevant.

3 Methodology

The main part of this thesis is the experimentation that lead to the proposed method, and the demonstration of its behaviour. To answer RQ1, a method is proposed, based on the assumption that promoting proportionality naturally increases diversity. A case based study answers RQ2, by demonstrating the strengths and weaknesses of this method.

This section starts off by describing the proposed method on a high level. After that the data used for testing is introduced, followed by a more detailed explanation of the implementation of the method. Finally the case based experiment is laid out in more detail.

3.1 Proposed method

The main inspiration for this method is PM-2 [12] and DDF [21] which is an extension of the former. These methods utilize Sainte-Laguë as a means to achieve proportional representation in regards to predefined subtopics of the data. Another inspiration is the method presented by Vee et al. [29], that diversify results based on different fields of structured data. The main contribution of this thesis is to present a novel method that combines these two ideas by using the Sainte-Laguë method on different rankings created by searching for the same query on different fields of structured data.

Lets say you are searching in a database of music and use the search query "rock", one could assume that the results would be a list of rock albums and rock bands. If the search engine is based on TF-IDF or something similar, you might just as well end up with a list of artists or bands with "rock" in their names. In either of these cases there is a glaring lack of diversity. Now lets say we boost any item where the search term hits in the genre field. This would naturally result in the former example. Conversely, if items with the term hitting in the name field are boosted, the list would end up as the latter example. To increase diversity we could select the top ranking from each field and repeat until the list is full. Than we would end up with half from each example, given that name and genre are the only searchable fields. One problem with this approach is that if we search for something with very little relevance to one of the fields, the result would still end up with quite a few items from that field and hence reducing the overall relevance of the result set. Going a step further we could instead put a penalty on the search scores for each field that is used to boost an item. This way we end up with a list of the top ranked from each field while preserving a fair amount of relevance.

This sounds a lot like the Sainte-Laguë method, discussed earlier. This method uses a quotation formula, based on the vote total, to determine which party should fill a given seat. The quotation formula is $\frac{V}{2s+1}$, where V is the vote total and s is the number of seats the party has already been allocated. This means that the divisor will first be 1, than 3, 5, 7 and so on. The quotation is recalculated for each party that receives a seat and than repeat the process until all seats are filled.

To apply the Sainte-Laguë method to structured data, the various fields of the data represents the political parties, spots in the final search result represents the seats of the parliament and the search score represents the vote totals. Search queries are run individually on each of the fields to create a set of initial rankings. Top position in the result set is given to the item with the overall highest score. The score for the next item in that ranking is then penalised using the quotation formula. A new item is then selected, with a reduced likelihood of coming from the same field as the previous one. This process is repeated until the result set is full.

Proportional representation of the relevance of a query on different fields, may not make a whole lot of sense, but the method may still benefit from increased diversity from the use of the Sainte-Laguë method, while potentially sacrificing some relevance.

3.2 Test data

For testing the method, data from Discogs[3] was used. Discogs is a community based site where anyone can add and edit data related to musical releases, be it CDs, vinyl reconds or digital releases, among others. The data has four major data types including artists, labels, releases and masters. Artists are solo artists, bands, band members and so on. They have fields including name and description, as well as references to other artists via alias, group, and member. Labels are recording companies and include fields for name, description and references to sub labels or parent labels. Releases are individual CDs, LPs, digital releases, and so on. Masters represent collections of releases of the same album or single etc. Their fields include title, genre, year among others. For convenience the individual releases and

[3] `https://www.discogs.com/`

Data available at: `https://data.discogs.com/`

labels have been omitted since they introduced relatively little new information compared to the amount of work it would take to include them.

In the data set there are about 6.6 million artists and 1.6 million master releases. The reason behind the higher number of artists is in part because of a lot of albums and singles having only one version registered on Discogs, and therefor do not have a master release. There are also many artists that have a very large number of aliases that are registered as individual artists. Quite a few of these artists have no master releases and some even have different aliases on the same album.

3.3 Implementation

The data is stored in OntoText's GraphDB[4] and fetched using Lucene. GraphDB Lucene ranks items using a variant of TF-IDF[5]. Queries are performed through a Lucene connector. These are indexes where it is possible to define more complex fields using property chains or combine fields into larger ones. This allows for querying entities based on fields of other entities that are connected to them. For example master releases can be retrieved based on the name of the artist, or name of other albums the artist has released.

For this experiment the chains are kept fairly shallow and fields are combined mainly in order to reduce the amount of queries that are sent to the database. Fields that are fairly similar or convey the same type of data are combined. Table 1 list the defined fields. In circumstances where the scores of the top item from several fields are the same, the ordering of the fields will affect the outcome of the final ranking. This is the same principle as the diversity ordering presented by Vee et al.[29]. The order that was used in this experiment is shown in the priority column of table 1

The initial ranking and rankings on the individual fields are based purely on default Lucene scoring, but could just as easily be applied to any type of ranking. After retrieving the rankings via the web API, the query combination using the Sainte-Laguë method is performed using a Python script. A pseudocode snippet of this is shown below.

[4] http://graphdb.ontotext.com/

[5] https://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

Field	Raw data fields	Priority
artist_name	artist > name artist > realname artist > namevariations	3
artist_group	artist > group-name artist > group > namevariations	6
artist_member	artist > member > name artist > member > namevariation	7
master_title	master > title	4
master_year	master > year	1
master_genre	master > genre master > style	2
master_artist_name	master > artist > name master > artist > realname master > artist > namevariation	5

Table 1: Fields that are used to create the initial rankings.

Pseudocode:

```
input: initial_rankings, score_caps, score_multipliers

final_ranking = []
sl_ratios = [1] * length(initial_rankings)

function calculate_score(item):
    score = item.get_score()
    score.apply_modifiers(score_caps, score_multipliers)
    item.sl_score = score / (2 * ratio + 1)

for each result in initial_rankings.get_top_results():
    calculate_score(result)

for each spot in final_ranking:
    field = highest_scoring_field()
    final_ranking.add(field.pop(0))
    sl_ratios[field] ++
    calculate_score(field[0])

return final_ranking
```

initial_rankings from the input are the rankings generated by querying different fields of the structured data. score_caps and score_multipliers are sets of modifiers that can be tweaked in order to mitigate the effect of uneven scoring of the initial rankings. The calculate_score function applies the score caps, score multipliers and the Sainte-Laguë formula to the top ranking item for each initial ranking. Scores for all top results are calculated prior to selecting the first result.

For each spot in the final result set, the method compares the scores from each top ranking item and adds the highest scoring to the final result. The highest_scoring_field() method simply returns the ranking that contains the highest scoring item. This item is then removed from its initial ranking and added to the final result. When an item is removed from one of the initial rankings and added to the result set, the score for the next item in the ranking is calculated, based on the increased Sainte-Laguë ratio.

Score caps can be dynamically determined by selecting a value based on the scores of the initial rankings before applying any modifiers. The final version of the method uses dynamic caps that are set to be equal to the highest scoring item. This is done in order to prevent any field from being boosted way higher than all the other fields and get over-represented in the final result.

3.4 Experiment

There has been little focus on diversifying search results on structured data, and there are no evaluation metrics in widespread use. Therefor it is not very useful to perform offline evaluation, since there's not much to compare it to. Instead a case based approach to evaluation was chosen, in order to shed light on some of the behaviour of the system, and findings are compared to those of previous studies. One of the main strengths of this approach is that it shows the actual behaviour of the system, instead of supplying numbers that do not always reflect user satisfaction. This can be very useful for further development of the approach, by highlighting behavioral traits that can be expanded on or needs to be addressed.

The main goal of this evaluation experiment, as stated in RQ1, is to show that this approach of query combination based on searches on different fields in structured data will increase diversity in the final result. Another goal is to show that there are values for the caps and multipliers that will ensure representation from all searches on fields that have at least one relevant result.

The final result of the method is compared to a ranking using a basic search engine. The search engine is the same Lucene search that is used for the initial rankings but is performed on all the fields combined into one. A few iterations of the method, using different constants, are tested using the same set of queries. The results are than presented in an incremental manner, where each of the modifiers are applied in sequence. First with just the basic Sainte-Laguë method, secondly by applying multipliers, thirdly by setting score caps, and finally a combination of both multipliers and caps. This presentation answers RQ2.

The queries used are a predefined set of varied terms that may or may not relate to music. These include artist names, genres, full and partial album names, years and a range of arbitrary words. Most of the queries were systematically selected to cover a variety of aspects, while others were selected randomly. Some of the queries are single-term, and others are multi-term. There are also a few queries that are combinations of multiple other queries, like an artists name and a year. The rarity of the query terms are also varied, where some of them appear very often, and others appear only a few times. Only a handful of the queries are presented, but they are selected in order to show patterns that seems to be consistent in the rest of the queries, and more are included in appendix A.

4 Case study

For some queries the benefit of this method is very clear. Take for instance the query "Bob Dylan". The search results for the initial search are shown in table 2. These are the results of combining all the fields into one field and performing a Lucene search with all the standard settings.

Rank	Score	Item	Type and ID
1	46.857	Bob Dylan & Friends	artist/3172917
2	42.605	Bob Dylan & His ...	artist/2512672
3	41.99	Bob Dylan	master/1235174
4	41.99	Bob Dylan	master/1514588
5	41.99	Bob Dylan	master/1068149
6	41.99	Bob Dylan	master/898747
7	41.99	Bob Dylan	master/4221
8	41.99	Bob Dylan	master/1404630
9	41.99	Bob Dylan	master/339724
10	41.99	Bob Dylan	master/992164

Table 2: Initial search results for "Bob Dylan"

It is not very surprising that the result set is filled with master releases. There has been released quite a few albums simply named Bob Dylan through his long career, most of which are compilation albums. It is quite obvious that a result set filled with mostly compilation albums of the same artist is not preferable, from a diversity standpoint.

The initial rankings used by the Sainte-Laguë method are retrieved in the same manner as in table 2, but performed on a single field. The Score column for all subsequent tables list the scores given to the items by Lucene for the search on these fields, before any modifiers and the Sainte-Laguë formula are applied. The field that was searched is listed under the From column. 'Group' means that the artist listed is in a group that matches the query, 'Title' means that the album title matches, and so on. Init Rank refers to the items position in the initial search results, performed on all fields put together. The Type and ID are Discogs own identifiers of the items and they provide a link to their respective pages on discogs.com.

4.1 Basic Sainte-Laguë

Table 3 shows the results after applying the Sainte-Laguë method with no caps or multipliers on the initial score.

Rank	Score	Init Rank	Item	Type and ID	From
1	11.919	603	George Recile	artist/465327	Group
2	10.844	72	Decades Live... ...	master/1620875	Artist
3	9.56	3	Bob Dylan	master/1235174	Title
4	8.64	1	Bob Dylan & Friends	artist/3172917	Member
5	10.81	73	Blind Boy Grunt	master/401131	Artist
6	10.556	624	Denny Freeman	artist/328687	Group
7	9.56	4	Bob Dylan	master/1514588	Title
8	8.64	591	Robert Zimmer An ...	artist/5571417	Member
9	10.81	74	Blackbushe	master/522473	Artist
10	10.447	626	Donnie Herron	artist/597967	Group

Table 3: Basic Sainte-Laguë for "Bob Dylan"

It is pretty clear that this ranking has a more diverse set of items. There is now a fairly even distribution of artists and releases. There is still a decent representation of Bob Dylans albums, but now there are a couple of them that are not named Bob Dylan. Artists also have different reasons for being included, where some of them are in groups with 'Bob Dylan' in their names and others are groups with Bob Dylan as a member.

The initial ranks mostly tent towards the top, but the artists that are selected using the Group field rank quite a bit lower. They are clearly still relevant since they are in Bob Dylan & His Band.

4.2 Sainte-Laguë with multipliers

Searching for something that would intuitively fit into one of the categories that tend to score lower, need a bit more modification than 'Bob Dylan'. Take for instance the query "rock". This, of course, is a music genre and it would be preferred to include at least some rock albums in the result. There is of course also artists and albums that have the word rock in them, but does not fit the rock genre, so a list full of only rock albums may be equally undesirable as no rock albums.

Rank	Score	Init Rank	Item	Type and ID	From
1	5.392	209	Rocket (4)	artist/249250	Member
2	5.197	341	I Can't Believe ...	master/84687	Artist
3	5.106	483	Rock Rock	artist/3565892	Name
4	4.761	>5000	Tiit Kõrvits	artist/2388313	Group
5	3.759	>5000	Rock, Rock, Rock	master/1261845	Title
6	5.189	346	Soul Survivor	master/39210	Artist
7	5.123	445	Soundcraft	artist/4659	Member
8	4.698	940	Hopey Rock	artist/523761	Name
9	4.689	>5000	Chris Brown (14)	artist/827561	Group
10	3.759	>5000	Rock Rock Rock	master/1343768	Title

Table 4: Basic Sainte-Laguë for "rock"

The initial scoring based purely on Lucene with all fields combined lists only artists. Some of them have released rock albums, but that is purely incidental. Applying the basic Sainte-Laguë method does help a lot with the diversity as shown in table 4. There is however a glaring lack of albums selected using the Genre field. This is due to scores for genres being very low compared to other fields. In fact the highest scoring item for the query 'rock' is lower than 0.8, while for other field it is way higher.

Rank	Score	Init Rank	Item	Type and ID	From
1	5.392	209	Rocket (4)	artist/249250	Member
2	5.197	341	I Can't Believe ...	master/84687	Artist
3	5.106	483	Rock Rock	artist/3565892	Name
4	4.761	>5000	Tiit Kõrvits	artist/2388313	Group
5	3.759	>5000	Rock, Rock, Rock	master/1261845	Title
6	5.189	346	Soul Survivor	master/39210	Artist
7	5.123	445	Soundcraft	artist/4659	Member
8	0.762	>5000	Alone	master/630593	Genre
9	4.698	940	Hopey Rock	artist/523761	Name
10	4.689	>5000	Chris Brown (14)	artist/827561	Group

Table 5: Sainte-Laguë with multiplier of 4 on genre, for "rock"

One obvious solution to this problem would be to boost the scores of low scoring fields so they are able to compete for the top ranks. Table 5 shows the results after multiplying the scores for the Genre field by 4, prior to applying the Sainte-Laguë formula. Now there is one album included that is selected using the Genre field. It is still not quite the desired result set, since there is only one item from genre, and it is ranked almost at the bottom, at 8th place.

Giving the field an even bigger boost should solve this problem. A multiplier of 8 seems to give the desired outcome, as shown in table 6.

Rank	Score	Init Rank	Item	Type and ID	From
1	0.762	>5000	Alone	master/630593	Genre
2	5.392	209	Rocket (4)	artist/249250	Member
3	5.197	341	I Can't Believe ...	master/84687	Artist
4	5.106	483	Rock Rock	artist/3565892	Name
5	4.761	>5000	Tiit Kõrvits	artist/2388313	Group
6	3.759	>5000	Rock, Rock, Rock	master/1261845	Title
7	0.762	>5000	Absolute Rock Ba ...	master/1046105	Genre
8	5.189	346	Soul Survivor	master/39210	Artist
9	5.123	445	Soundcraft	artist/4659	Member
10	4.698	940	Hopey Rock	artist/523761	Name

Table 6: Sainte-Laguë with multiplier of 8 on genre, for "rock"

Now there are two items selected using the Genre field, which is the ideal number for this query, given that there are no other fields represented by 3 or more items.

Rock is of course a very common genre and should be considered by Lucene to be of low importance when querying the Genre field. So lets take a look at a style (included in genre) that is not very common, namely shoegaze. In the data set there are just over 5,000 master

Rank	Score	Init Rank	Item	Type and ID	From
1	3.287	14	Stargaze	master/1509549	Genre
2	3.287	15	Closer	master/1509554	Genre
3	3.287	16	Colours & Changes	master/913533	Genre
4	9.246	1	Voidhangers	master/1332102	Artist
5	8.785	2	Shoegaze	artist/3397078	Name
6	3.287	17	Goldenwest	master/1178926	Genre
7	8.553	8	Pascal Riffaud	artist/3460589	Group
8	3.287	18	Grandfeathered	master/963599	Genre
9	6.94	13	Shoegaze EP	master/355808	Title
10	3.287	19	Sundays	master/1436980	Genre

Table 7: Sainte-Laguë with multiplier of 8 on genre, for "shoegaze"

releases marked as shoegaze, while rock is the most common genre with more than 500,000 master releases. Since the initial rankings use TF-IDF it would be fair to assume that 'shoegaze' is considered a more important term, and should result in higher scores.

Table 7 shows the results after using the same constants that was used for rock in table 6. More than half of the results are now chosen from the Genre ranking. This is clearly not very good for diversity. It is not possible to solve this problem with multipliers alone. The multiplier will either be set too low and include too few results from the Genre field when searching for common genres or too many when searching for rare genres.

4.3 Sainte-Laguë with score caps

A possible solution to the problem of boosting some terms too much would be to set a cap on scores. Lets first examine what happens when applying caps without boosting any of the fields, to see if this can be used alone, to solve the diversity problem.

The Year field has some of the same properties as the Genre field, so lets take a look at the query '1975'. For this query it would make sense to include albums that were released in 1975, but also artists and albums that include 1975 in their names and titles.

Rank	Score	Init Rank	Item	Type and ID	From
1	8.027	2	TOOTIMETOOTIMETO ...	master/1440438	Artist
2	7.924	39	Matthew Healy	artist/3480219	Group
3	6.649	32	The 1975	artist/3231113	Name
4	5.267	91	1975	master/1499055	Title
5	8.027	1	The 1975	master/596580	Artist
6	7.924	40	Adam Hann	artist/3480220	Group
7	6.649	33	Model 1975	artist/1942854	Name
8	8.027	3	I Like It When Y ...	master/963890	Artist
9	7.924	41	Ross MacDonald (2)	artist/3480221	Group
10	5.267	92	1975	master/567145	Title

Table 8: Basic Sainte-Laguë for "1975"

In the top 10 results of the basic Lucene search on all fields, there are only releases, but none of them were released in 1975. Results for performing the basic Sainte-Laguë search are listed in table 8. The improvements of the results are comparable to the ones for the previous queries.

There is a fairly even distribution of artist and masters, that come from a variety of initial rankings. However, once again, there is a lack of inclusions of the most obvious items, namely albums released in 1975.

Applying caps to the initial scores are meant to reduce the likelihood of flooding the results with items from fields with very high scores.

In table 9 the initial scores have been capped at 7, which is just a little bit lower than the highest scoring items. The internal ranking in the fields have not been changed.

Rank	Score	Init Rank	Item	Type and ID	From
1	8.027	2	TOOTIMETOOTIMETO ...	master/1440438	Artist
2	7.924	39	Matthew Healy	artist/3480219	Group
3	6.649	32	The 1975	artist/3231113	Name
4	5.267	91	1975	master/1499055	Title
5	8.027	1	The 1975	master/596580	Artist
6	7.924	40	Adam Hann	artist/3480220	Group
7	6.649	33	Model 1975	artist/1942854	Name
8	5.267	92	1975	master/567145	Title
9	8.027	3	I Like It When Y ...	master/963890	Artist
10	7.924	41	Ross MacDonald (2)	artist/3480221	Group

Table 9: Sainte-Laguë with cap of 7, for "1975"

When applying the cap of 7 the result set is almost unchanged. The only difference is a shuffling of the three lowest ranked items. Since the cap is lower than the highest ranking results from several of the initial rankings, the ordering of the fields come into play. Now all the results are included in the priority ordering of the fields, but there is still a lack of albums released in 1975.

In order for any releases from 1975 to be included the cap has to be set significantly lower. Setting the cap to 2 ensure inclusions of albums selected using the Year field, as shown in table 10. For this query this seems to defeat some of the purpose of applying the Sainte-Laguë method, since the top 10 results are now selected sequentially from each field in the defined order, basically ignoring the score.

What the cap essentially does is ensure that any rankings with scores above or within a certain range below the cap will have some representation in the final ranking, given that the number

Rank	Score	Init Rank	Item	Type and ID	From
1	2.075	506	Beethoven "Emper ...	master/1541036	Year
2	6.649	32	The 1975	artist/3231113	Name
3	5.267	91	1975	master/1499055	Title
4	8.027	2	TOOTIMETOOTIMETO ...	master/1440438	Artist
5	7.924	39	Matthew Healy	artist/3480219	Group
6	2.075	507	Clay & Vicki Ca ...	master/1544074	Year
7	6.649	33	Model 1975	artist/1942854	Name
8	5.267	92	1975	master/567145	Title
9	8.027	1	The 1975	master/596580	Artist
10	7.924	40	Adam Hann	artist/3480220	Group

Table 10: Sainte-Laguë with cap of 2, for "1975"

of fields is lower than the number of included results. This does on its own ensure diversity in the final result, and prevent any field from being overrepresented, but it defeats the purpose of proportional representation and reduces the method down to only taking rank into account.

4.4 Combining caps and multipliers

The final version of the method is to use both caps and multipliers. A cap can be used to avoid getting very high scores when multiplying some fields, and filling the list with items ranked on these fields. For each query that has been tested during this experiment the scores for the top results have been fairly even, with the exception of Year and Genre, which have been consistently lower than the rest. However the scores have fluctuated quite a bit between the different queries. Usually the scores are somewhere between 4 and 9, but sometimes go a bit lower and occasionally go as high as 20. This means that setting a constant cap for all queries would result in the cap being too high for some queries, or too low for others.

A solution to this problem is to set a dynamic cap. For this method it is done by simply setting the cap to be the same as the highest score in any of the initial rankings. This will ensure that no field gets boosted higher than the initial highest score, which could result in that field taking too many of the slots in the final result.

The only two fields that consistently need a boost to be competitive among the other fields are year and genre. The Lucene ranking used for the initial rankings does not allow for partial term matches. This means that if the query matches the Year it most certainly came out that year, or at least is indicated as it on Discogs. There are some genres with multiple words in their names, so it is not quite the same story for matches on the Genre field, but close enough for this experiment. For the rest of this thesis, the method used will therefore use a very high multiplier for the Year and Genre field, essentially ignoring the scores and using only the rank for these two fields. All other fields will have no multiplier.

Lets take a look at some special cases to see if the results are satisfactory. More results can be found in appendix A.

Rank	Score	Init Rank	Item	Type and ID	From
1	7.372	2	Rapoon	artist/7029	Name
2	7.259	3	Darker By Light	master/1304982	Artist
3	7.259	4	Alien Glyph Morp ...	master/38950	Artist
4	4.26	1	Rapoon Live At L ...	master/1307882	Title
5	7.259	5	Rhiz	master/1169989	Artist
6	7.259	6	Airstrikes	master/1257352	Artist
7	7.259	7	Jane From Whitle ...	master/1162577	Artist
8	7.259	8	My Life As A Ghost	master/1170007	Artist
9	7.259	9	Andre On The Line	master/1237900	Artist
10	7.259	10	In Time Past	master/1301561	Artist

Table 11: Query: rapoon, Cap: dynamic, Multipliers: 10 for year and genre

The first query is 'rapoon' and the results are listed in table 11. This query matches only one artist and the name of one of his albums. All other results are releases by this artist. The results are obviously about as perfect as they can be, and have both items that differ from the rest, near the top.

The second case is a query with several terms. Table 12 shows the results from searching for the band Rage Against the Machine. This should obviously include items that are relevant to the band. Examining the items in the ranking reveals that all but three of the items are directly related to the band, either as an album they released or members of the band.

Two of the three items that are not directly related have names that are most likely parodying the name, by simply adding an extra word. Rage Against the Machine should probably be ranked above these two, but that would require a different method for the initial rankings.

Rank	Score	Init Rank	Item	Type and ID	From
1	19.157	18	Renegades	master/7877	Artist
2	15.616	73	Tim Commerford	artist/420215	Group
3	11.647	66	Rage Against The Brexit Ma...	artist/6853704	Name
4	19.157	1	Rage Against The Machine	master/7939	Artist
5	15.194	74	Brad Wilk	artist/332379	Group
6	8.639	107	NERVOUS//CORPSE	artist/5238772	Member
7	19.157	19	Evil Empire	master/7951	Artist
8	11.647	67	Rage Against The Wash Mac...	artist/6733042	Name
9	19.157	20	MP3	master/1136311	Artist
10	14.706	75	Zack de La Rocha	artist/64348	Group

Table 12: Query: rage against the machine, Cap: dynamic, Multipliers: 10 for year and genre

The last item is the only one chosen using the Member field and matches only on the word 'rage'. There is an argument to be made for not including this item. Tweaking the system to avoid including such items is left for future work.

From a pure diversity perspective this result set is pretty good, where all but one of the results are obviously relevant.

Now lets try searching for only the word 'against'. There are several bands that use against in their names so it would be good to have a decent representation of them in the result set. Table 13 show the results for this query.

Rank	Score	Init Rank	Item	Type and ID	From
1	7.634	6	Black Me Out	master/719883	Artist
2	6.29	148	Against	artist/1425539	Name
3	6.249	585	James Bowman	artist/553943	Group
4	5.041	1117	Against	master/588859	Title
5	4.979	1	Against The Current	artist/4444059	Member
6	7.634	7	Stabitha Christie	master/1458804	Artist
7	6.249	586	Dustin Fridkin	artist/1280181	Group
8	6.157	187	Against Me!	artist/253797	Name
9	7.634	8	As The Eternal C ...	master/52276	Artist
10	5.041	1118	Against	master/57979	Title

Table 13: Query: against, Cap: dynamic, Multipliers: 10 for year and genre

Three bands with the term 'against' in their title are represented with at least one item.

Against Me! is represented by 6 items; 3 releases, 2 band members and the band itself. This is another problem that is caused by the underlying ranking algorithm, and may be addressed by using a different underlying ranking method or using additional diversification techniques.

The last case that is examined is combining several unrelated terms. The query in question is 'rapoon 1975 decemberunderground', and the results are shown in table 14. Results for 'rapoon' and '1975' were presented and discussed earlier. They were chosen because of their very few and many query matches, respectively. The last term is one that returns only one result, namely an album with that title.

Rank	Score	Init Rank	Item	Type and ID	From
1	2.075	583	Beethoven "Emper ...	master/1541036	Year
2	8.99	144	Decemberunderground	master/32617	Title
3	8.027	3	TOOTIMETOOTIMETO ...	master/1440438	Artist
4	7.924	115	Matthew Healy	artist/3480219	Group
5	7.372	24	Rapoon	artist/7029	Name
6	2.075	584	Clay & Vicki Ca ...	master/1544074	Year
7	8.027	1	The 1975	master/596580	Artist
8	7.924	116	Adam Hann	artist/3480220	Group
9	6.649	104	The 1975	artist/3231113	Name
10	2.075	585	Elton John Rock Hits	master/353939	Year

Table 14: Query: rapoon 1975 decemberunderground, Cap: dynamic, Multipliers: 10 for year and genre

The Decemberunderground album is listed at rank 2. Rapoon is reprecented by only one item, which is the artist itself. 1975 takes all the other slots.

These queries does highlight some problems with the system as it is now. Some of these problems are introduced by the system, and others are caused by the chosen method for generating the initial rankings. From a purely diversity perspective these results are still far better than purely going by the basic Lucene ranking. This shows that the query combination algorithm utilizing the Sainte-Laguë method does in fact promote diverse results, while still keeping most of the relevance.

5 Summary

This thesis is an investigation into the possibilities of diversifying information retrieval on structured data. The main contribution is a proposed method and a case study that show some of its behaviour. To reiterate, the research questions are the following:

RQ1: What method can be used to improve diversity in IR on structured data?

RQ2: How does the chosen method behave?

The thesis started off with an overview of the various techniques used in order to improve information retrieval and filtering in relation to metrics that go beyond the traditional accuracy based measures. A query combination method using the Sainte-Laguë method to promote proportionality was presented, followed by a case based study on its behavioral strengths and weaknesses. This chapter will summarize the thesis and findings of the case study.

Goals of diversity in IR are resolving ambiguous queries and avoiding duplicity. In previous research there has been a range of different approaches to solving these problems. One of the techniques that has been successful, is diversifying in relation to explicitly assigned subtopics on non-structured data. Promoting proportionality using the Sainte-Laguë method naturally improves diversity.

The chosen method for this thesis has taken inspiration from research on subtopics, and use the different fields of structured data as a way of imitating subtopics, without having to explicitly assign them. The method uses query combination to merge the rankings of search queries performed on different fields of the same data. Items are selected using the Sainte-Laguë method to promote proportional representation of the fields, ensuring diversity. The goal of using proportionality is to include items that score high for a given field, but excluding items with low scores on the same field. Two modifiers were defined, in order to mitigate some of the problems faced when simply applying the basic Sainte-Laguë method. A multiplier for each field boosts fields that in general score lower than other fields, and a score cap prevents fields from being overrepresented, if their initial score is very high.

A case based study was conducted as a proof of concept on the chosen method, and have shown that it is a viable option. The study investigated different iterations of the method, in turn applying the modifiers on their own, and finally combining them into the final proposed method.

For most queries the basic Sainte-Laguë method is enough to achieve satisfactory diversity in the search results. However for searches that should match well on the Genre and Year fields, there was a glaring lack of items from these fields. Boosting these fields does ensure that they are represented, but when querying rare terms, too many items from these fields were included. Applying caps to the initial scores ensures representation from fields that have scores higher or very close to the cap. This ensures diversity of the result set, but ends up defeating some of the purpose of using proportional representation. It reduces the method down to only considering the rank, and not the scores, from the initial rankings. The final method applies both multipliers and cap at the same time eliminates these two problems. A dynamic cap that is set to the highest scoring item from any of the initial rankings, prevents the multipliers from boosting any fields too high.

The multipliers did not prove useful for this particular experiment, other than to transform the Year and Genre fields into only considering the rank of the items. These two fields have, for the most part, significantly lower scores than the rest, requiring a boost to make them competitive. However, for some queries they had very high scores, and didn't need boosting. If a query matches any of these fields it means that it is almost certainly relevant for it. This leads to the easiest solution being simply boosting those fields so high that score becomes irrelevant. This does not mean that the multipliers are useless in all scenarios. In fact they may even be useful for the scenario used in this experiment, but it would require a different type of study to verify the usefulness. The title field often gets slightly lower scores than the other fields and may benefit from a slight boost. Doing such minor adjustments requires a lot more comprehensive testing, which is outside the scope of this thesis.

Testing on the final method show that it does in fact provide increased diversity compared to the underlying scoring method. All tested queries show improvements from a diversity standpoint. These findings are consistent with the idea that data fusion and query combination increases diversity [4]. Ultimately the method has shown great promise as a viable option for diversifying IR on structured data.

6 Future work

While the method in itself has been shown effective at its goal, there are still many areas of improvement. The method should also be more thoroughly evaluated to verify its usefulness. This section gives an overview of some of the directions that can be taken by future work.

A natural next step for this research is to conduct online evaluation, to compare user satisfaction of this system, to other diversity approaches.

For rankings using this method, it will not always be immediately obvious why an item was added to the result set. In order to maximize user satisfaction, explanations for some of the results would be crucial. For instance when there is a match on a band name, and a member is listed in the results, it would be a good idea to display the band name as well. There is a fairly straight forward way to explain why the result is relevant, by simply mapping each initial ranking to an explanation, that would be displayed beside it.

When a query is less ambiguous, or the ambiguity lies within one field, the result set will still end up with many similar items. For instance when searching for Bob Dylan (table 3), the band members that are listed are all from the same band, even tho there are other bands with 'Bob Dylan' in their names. A possible way of including some more diversity in these circumstances would be to have some more elaborate field definitions that are utilized only when there is a lack of diversity without them. One such field could be bands that members of a matching band are also in. So when searching for Rapoon (table 11), which is the project of Robin Storey, who is also a past member of the group Zoviet France. Listing Zoviet French and an album of theirs could be beneficial.

Another possible solution to this problem could be to diversify the initial rankings, before combining them with the Sainte-Laguë method. Applying something like the greedy reranking method of Chen and Karger [10], could do the trick. This would of course add another layer of complexity, that would make it run even slower, but could be worth looking into.

One fairly obvious problem of this approach is that the collection has to be queried several times in order to create the initial rankings that are later combined into the final result. The severity of this may vary greatly depending on the database system in use. There is a trade-off between diversity and speed. If more fields are defined, the final result will be more diverse, but it will take longer to execute. Measuring this effect, minimizing latency and finding a good trade-off, between diversity and speed, is left for future work.

Using a ranking method, other than TF-IDF, may mitigate some of the problems of some terms being considered way more important than others. Perhaps a method like BM25 could work well. As was done in the final method, using different ranking methods for some fields, may also be beneficial. The year field for instance, using TF-IDF caused more harm than good. Since the amount of albums being released is increasing each year, searching for a recent year would mean that the initial scores for the year column are quite a bit lower than if searching for one of the earliest years, which has very few releases. This is due to the nature of TF-IDF where the term 2019 appears a lot more often than the term 1900 and hence the latter is considered more important.

There are alternative methods to the Sainte-Laguë method, for obtaining proportional representation, that are used in political elections. Some of which favor smaller parties and others favor larger parties. It would be interesting to compare results from using some of these alternatives. Perhaps using a method that favors small parties even more would mitigate the need to compensate for the low scores in certain fields, and promote more novel results.

One thing that was not considered in this thesis is how introducing popularity of items would affect the ability for this method. Popular items could get a boost in the initial rankings, and therefore be more likely to end up in the final result list. This can be done by applying data fusing on this method and a ranking based on popularity of the items. Intuitively this should not cause any major problems, but should be verified in practise.

Using machine learning at different stages, could be a good way to tweak the system. It can be used for learning good ratios for the different fields in the Sainte-Laguë method, by fx. measuring relevance for the top results, to make sure there are no completely irrelevant results because some of the fields get priority. Machine learning can also be used in online evaluation by allowing people to rate how well they think results match their information need.

Bibliography

[1] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, page 5–14, New York, NY, USA, 2009. Association for Computing Machinery.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval (Second edition)*. New York: ACM Press; Harlow, England: Addison-Wesley, 2011.

[3] Krisztian Balog. *Entity-Oriented Search*. Springer Publishing Company, Incorporated, 1st edition, 2018.

[4] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448, May 1995.

[5] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38, December 1992.

[6] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, page 25–32, New York, NY, USA, 2004. Association for Computing Machinery.

[7] Fei Cai, Ridho Reinanda, and Maarten De Rijke. Diversifying query auto-completion. *ACM Trans. Inf. Syst.*, 34(4), June 2016.

[8] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA, 1998. Association for Computing Machinery.

[9] Sylvain Castagnos, Armelle Brun, and Anne Boyer. When Diversity Is Needed... But Not Expected! In *International Conference on Advances in Information Mining and Management*, pages 44–50, Lisbon, Portugal, November 2013. IARIA XPS Press.

[10] Harr Chen and David R. Karger. Less is more: Probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, page 429–436, New York, NY, USA, 2006. Association for Computing Machinery.

[11] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 659–666, New York, NY, USA, 2008. Association for Computing Machinery.

[12] Van Dang and W. Bruce Croft. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, page 65–74, New York, NY, USA, 2012. Association for Computing Machinery.

[13] Michael Gallagher. Proportionality, disproportionality and electoral system. *Electoral studies*, 10(1):33–51, 1991.

[14] William Goffman. A searching procedure for information retrieval. *Information Storage and Retrieval*, 2(2):73–78, 1964.

[15] Uri Hanani, Bracha Shapira, and Peretz Shoval. Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203–259, August 2001.

[16] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[17] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[18] Marius Kaminskas and Derek Bridge. Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Trans. Interact. Intell. Syst.*, 7(1), December 2016.

[19] Komal Kapoor, Vikas Kumar, Loren Terveen, Joseph A. Konstan, and Paul Schrater. "i like to explore sometimes": Adapting to dynamic user novelty preferences. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 19–26, New York, NY, USA, 2015. Association for Computing Machinery.

[20] Neal Lathia, Stephen Hailes, Licia Capra, and Xavier Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, page 210–217, New York, NY, USA, 2010. Association for Computing Machinery.

[21] Shangsong Liang, Zhaochun Ren, and Maarten de Rijke. Fusion helps diversification. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, page 303–312, New York, NY, USA, 2014. Association for Computing Machinery.

[22] Andrii Maksai, Florent Garcin, and Boi Faltings. Predicting online performance of news recommender systems through richer evaluation metrics. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, page 179–186, New York, NY, USA, 2015. Association for Computing Machinery.

[23] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, page 1097–1101, New York, NY, USA, 2006. Association for Computing Machinery.

[24] Stephen E Robertson. The probability ranking principle in ir. *Journal of Documentation*, 1977.

[25] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[26] Stephen E Robertson, Hugo Zaragoza, and Michael Taylor. Simple bm25 extension to multiple weighted fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, page 42–49, New York, NY, USA, 2004. Association for Computing Machinery.

[27] Alan Said, Ben Fields, Brijnesh J. Jain, and Sahin Albayrak. User-centric evaluation of a k-furthest neighbor collaborative filtering recommender algorithm. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, page 1399–1408, New York, NY, USA, 2013. Association for Computing Machinery.

[28] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, page 109–116, New York, NY, USA, 2011. Association for Computing Machinery.

[29] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. A. Yahia. Efficient computation of diverse query results. In *2008 IEEE 24th International Conference on Data Engineering*, pages 228–236, April 2008.

[30] Martijn C Willemsen, Bart P Knijnenburg, Mark P Graus, Linda CM Velter-Bremmers, and Kai Fu. Using latent features diversification to reduce choice difficulty in recommendation lists. *RecSys*, 11(2011):14–20, 2011.

[31] Cheng Xiang Zhai, William W. Cohen, and John Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, page 10–17, New York, NY, USA, 2003. Association for Computing Machinery.

[32] Hang Zhang and Vinay Setty. Finding diverse needles in a haystack of comments: Social media exploration for news. In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, page 286–290, New York, NY, USA, 2016. Association for Computing Machinery.

[33] Wei Zheng, Xuanhui Wang, Hui Fang, and Hong Cheng. Coverage-based search result diversification. *Inf. Retr.*, 15(5):433–457, October 2012.

[34] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515, 2010.

[35] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, page 22–32, New York, NY, USA, 2005. Association for Computing Machinery.

A Final ranking results

Rank	Score	Init Rank	Item	Type and ID	From
1	0.762	>5000	Alone	master/630593	Genre
2	5.392	209	Rocket (4)	artist/249250	Member
3	5.197	341	I Can't Believe ...	master/84687	Artist
4	5.106	483	Rock Rock	artist/3565892	Name
5	4.761	>5000	Tiit Kõrvits	artist/2388313	Group
6	3.759	>5000	Rock, Rock, Rock	master/1261845	Title
7	0.762	>5000	Absolute Rock Ba ...	master/1046105	Genre
8	5.189	346	Soul Survivor	master/39210	Artist
9	5.123	445	Soundcraft	artist/4659	Member
10	4.698	940	Hopey Rock	artist/523761	Name

Table 15: Query: rock

Rank	Score	Init Rank	Item	Type and ID	From
1	1.675	>5000	Minneapolis	master/416818	Genre
2	4.856	2084	Santa Guy Guyna	master/1204130	Artist
3	4.822	2	Jazz K 14	artist/3186650	Member
4	4.411	3072	Jazzy Jazz	artist/1667084	Name
5	3.689	>5000	Jazz Nocturno Jazz	master/958663	Title
6	3.524	>5000	DJ Smash	artist/8186	Group
7	1.675	>5000	Blue Note Meets ...	master/202313	Genre
8	4.856	2085	Timothee Abangi ...	master/1596964	Artist
9	4.822	2225	Swamp Click	artist/1219098	Member
10	4.411	3073	Jazz Black	artist/286918	Name

Table 16: Query: jazz

Rank	Score	Init Rank	Item	Type and ID	From
1	3.287	14	Stargaze	master/1509549	Genre
2	9.246	1	Voidhangers	master/1332102	Artist
3	8.785	2	Shoegaze	artist/3397078	Name
4	8.553	8	Pascal Riffaud	artist/3460589	Group
5	6.94	13	Shoegaze EP	master/355808	Title
6	3.287	15	Closer	master/1509554	Genre
7	8.287	9	Camille Michel	artist/4993227	Group
8	7.535	3	Extreme Shoegaze	artist/5688721	Name
9	3.287	16	Colours & Changes	master/913533	Genre
10	5.449	693	El Shoegaze Boss ...	master/1395451	Title

Table 17: Query: shoegaze

Rank	Score	Init Rank	Item	Type and ID	From
1	3.279	>5000	Lost Tree	master/1420811	Genre
2	8.643	9	On The Town	master/193860	Artist
3	7.43	1	Bill & Doree Post	artist/1640821	Member
4	7.232	5	Bill Post	artist/764633	Group
5	5.916	191	Karel Post	artist/618575	Name
6	5.809	4313	Nothing But... P ...	master/548490	Title
7	3.279	>5000	Parmi Les Autres	master/1519722	Genre
8	7.308	37	Friend And Lover	artist/132934	Member
9	7.232	4	Doree Post	artist/764634	Group
10	3.279	>5000	Espérer Sombrer	master/1525860	Genre

Table 18: Query: post rock

Rank	Score	Init Rank	Item	Type and ID	From
1	3.063	>5000	Starting A Fire	master/1368610	Genre
2	6.306	14	Xenon (3)	artist/230195	Member
3	5.73	30	Dance-Dance	artist/4100643	Name
4	5.136	104	Rap-O-Hush	master/280152	Artist
5	4.56	4254	Andrzej Zielińsk ...	artist/932856	Group
6	3.798	>5000	Dance, Dance, Dance	master/185465	Title
7	3.063	>5000	Fun Tonite	master/1368612	Genre
8	6.011	18	Andarecontrocorrente	artist/553266	Member
9	5.519	46	Dance Dance Dance	artist/5921634	Name
10	5.136	105	You Can Do It	master/112706	Artist

Table 19: Query: dance

Rank	Score	Init Rank	Item	Type and ID	From
1	2.278	>5000	The Blues (A Smi ...	master/1643050	Genre
2	6.212	1	Blues Brothers A ...	artist/439777	Member
3	5.398	96	Red Hot From Alex	master/344745	Artist
4	4.926	1827	"Blues"	artist/1052214	Name
5	4.4	>5000	Kenny Lamb	artist/762224	Group
6	3.841	>5000	Blues-Blues-Blues	master/878516	Title
7	2.275	>5000	The American Fol ...	master/1468916	Genre
8	6.005	661	Tomasz Krzemińsk ...	artist/5892569	Member
9	5.398	97	At The Cavern	master/558611	Artist
10	4.904	1888	Blues Baby Blues	artist/5460084	Name

Table 20: Query: blues

Rank	Score	Init Rank	Item	Type and ID	From
1	3.7	4979	No More Tears	master/233927	Genre
2	12.59	3	Horns In The House	master/1509871	Artist
3	9.944	300	Janusz Kowalski	artist/1370629	Group
4	9.747	34	Heavy Metal	artist/376088	Name
5	7.923	4862	Heavy Metal	master/1090908	Title
6	7.575	187	Apocalypse Babys	artist/1706247	Member
7	3.7	4980	Into The Future	master/1190228	Genre
8	11.954	7	Kitsch	master/290188	Artist
9	9.944	301	Jarosław Małys	artist/1370631	Group
10	3.7	4981	A Traves De Los ...	master/1388376	Genre

Table 21: Query: heavy metal

Rank	Score	Init Rank	Item	Type and ID	From
1	1.788	397	Super Mario Hist ...	master/294304	Year
2	7.71	3	Blinkgürtel	master/374470	Artist
3	7.394	9	2010	artist/2454279	Name
4	7.338	53	Mano Hetharia	artist/5158011	Group
5	6.335	27	Benno Kusche Mit ...	artist/7419697	Member
6	5.119	173	2010	master/845762	Title
7	1.788	779	Farewell (The Fi ...	master/1564697	Year
8	7.591	4	Heroes	master/1011435	Artist
9	7.338	54	Rosa van der Vijver	artist/5158015	Group
10	6.416	14	Juniors 2010	artist/6361584	Name

Table 22: Query: 2010

Rank	Score	Init Rank	Item	Type and ID	From
1	6.314	88	Old Bill Jones	master/1503573	Year
2	8.14	3	1900	artist/2002060	Name
3	7.925	4	Orgue Limonaire ...	master/882447	Artist
4	7.684	6	Tony Meler Et So ...	artist/2368689	Member
5	7.568	49	Marco Talini	artist/779428	Group
6	6.782	81	1900?	master/828107	Title
7	7.925	5	Les Plus Jolies ...	master/1517287	Artist
8	7.062	10	1900 Storm	artist/2848311	Name
9	6.782	82	1900	master/482370	Title
10	7.925	2	The "Limonaire 1 ...	master/1379626	Artist

Table 23: Query: 1900

Rank	Score	Init Rank	Item	Type and ID	From
1	5.252	13	One Sweetly Sole ...	master/1180753	Year
2	10.417	1	Euthanasie Records	artist/4132470	Member
3	10.076	2	Deliver Us	master/714545	Artist
4	8.996	4	David 1904	artist/3752384	Name
5	7.672	10	1904	master/1019826	Title
6	5.252	14	Washington Post ...	master/1266719	Year
7	9.454	3	Blood Red / Demo ...	master/1441521	Artist
8	8.877	5	1904	artist/1191965	Name
9	5.252	15	Queen Of Sheba March	master/1636581	Year
10	6.622	11	Elbing 1904	master/1356329	Title

Table 24: Query: 1904

Rank	Score	Init Rank	Item	Type and ID	From
1	2.075	506	Beethoven "Emper ...	master/1541036	Year
2	8.027	2	TOOTIMETOOTIMETO ...	master/1440438	Artist
3	7.924	39	Matthew Healy	artist/3480219	Group
4	6.649	32	The 1975	artist/3231113	Name
5	5.267	91	1975	master/1499055	Title
6	2.075	507	Clay & Vicki Ca ...	master/1544074	Year
7	8.027	1	The 1975	master/596580	Artist
8	7.924	40	Adam Hann	artist/3480220	Group
9	6.649	33	Model 1975	artist/1942854	Name
10	2.075	508	Elton John Rock Hits	master/353939	Year

Table 25: Query: 1975

Rank	Score	Init Rank	Item	Type and ID	From
1	11.919	603	George Recile	artist/465327	Group
2	10.844	72	Decades Live... ...	master/1620875	Artist
3	9.56	3	Bob Dylan	master/1235174	Title
4	8.64	1	Bob Dylan & Friends	artist/3172917	Member
5	10.81	73	Blind Boy Grunt	master/401131	Artist
6	10.556	624	Denny Freeman	artist/328687	Group
7	9.56	4	Bob Dylan	master/1514588	Title
8	8.64	591	Robert Zimmer An ...	artist/5571417	Member
9	10.81	74	Blackbushe	master/522473	Artist
10	10.447	626	Donnie Herron	artist/597967	Group

Table 26: Query: bob dylan

Rank	Score	Init Rank	Item	Type and ID	From
1	19.157	18	Renegades	master/7877	Artist
2	15.616	73	Tim Commerford	artist/420215	Group
3	11.647	66	Rage Against The ...	artist/6853704	Name
4	19.157	1	Rage Against The ...	master/7939	Artist
5	15.194	74	Brad Wilk	artist/332379	Group
6	8.639	107	NERVOUS//CORPSE	artist/5238772	Member
7	19.157	19	Evil Empire	master/7951	Artist
8	11.647	67	Rage Against The ...	artist/6733042	Name
9	19.157	20	MP3	master/1136311	Artist
10	14.706	75	Zack de La Rocha	artist/64348	Group

Table 27: Query: rage against the machine

Rank	Score	Init Rank	Item	Type and ID	From
1	9.949	1	C'mon Talk	master/683326	Artist
2	8.431	11	Alan Bernhoft	artist/2473888	Name
3	8.377	12	Explicit Lyrics	artist/2066421	Member
4	6.409	8	Sigbjørn Bernhof ...	master/1438288	Title
5	9.949	2	Stop/Shutup/Shou ...	master/1400925	Artist
6	8.319	13	Bernhoft	artist/4665809	Name
7	7.688	15	Span (2)	artist/288978	Member
8	9.949	3	Ceramik City Chr ...	master/710437	Artist
9	7.778	14	Jarle Bernhoft ...	artist/3837975	Name
10	9.949	4	1: Man 2: Band	master/315028	Artist

Table 28: Query: bernhoft

Rank	Score	Init Rank	Item	Type and ID	From
1	9.49	1	Grails	artist/251435	Name
2	8.8	2	The Burden Of Hope	master/26213	Artist
3	8.683	16	William Slater	artist/626084	Group
4	8.442	22	Grails	master/977990	Title
5	8.8	3	Take Refuge In C ...	master/6097	Artist
6	8.683	17	Timothy Horner	artist/648947	Group
7	8.14	15	Jeff Grails	artist/6532611	Name
8	7.286	23	Unholy Grails	master/1553885	Title
9	8.8	4	Burning Off Impu ...	master/6075	Artist
10	8.394	18	Ben Nugent	artist/626628	Group

Table 29: Query: grails

Rank	Score	Init Rank	Item	Type and ID	From
1	19.254	1	Hammer And Anvil	master/283515	Artist
2	16.597	14	Jamie Willcox	artist/883223	Group
3	13.731	10	Pure Reason Revo ...	artist/337983	Name
4	19.254	2	The Dark Third	master/165401	Artist
5	16.597	15	Paul Glover	artist/1134757	Group
6	19.254	3	The Intention Craft	master/165402	Artist
7	7.979	46	The Tinkers (3)	artist/4057960	Member
8	16.597	16	Greg Jong	artist/1619232	Group
9	19.254	4	Apprentice Of Th ...	master/237372	Artist
10	5.329	3028	Pure Pure	master/1450260	Title

Table 30: Query: pure reason revolution

Rank	Score	Init Rank	Item	Type and ID	From
1	7.372	2	Rapoon	artist/7029	Name
2	7.259	3	Darker By Light	master/1304982	Artist
3	7.259	4	Alien Glyph Morp ...	master/38950	Artist
4	4.26	1	Rapoon Live At L ...	master/1307882	Title
5	7.259	5	Rhiz	master/1169989	Artist
6	7.259	6	Airstrikes	master/1257352	Artist
7	7.259	7	Jane From Whitle ...	master/1162577	Artist
8	7.259	8	My Life As A Ghost	master/1170007	Artist
9	7.259	9	Andre On The Line	master/1237900	Artist
10	7.259	10	In Time Past	master/1301561	Artist

Table 31: Query: rapoon

Rank	Score	Init Rank	Item	Type and ID	From
1	2.739	>5000		master/1614455	Genre
2	12.484	4	Turtledove	artist/6910084	Member
3	11.997	2	This Machine Run ...	master/703100	Artist
4	10.74	8	Fear Of The Dark	artist/1657193	Name
5	9.018	1092	Sebastian Schoepsdau	artist/2561609	Group
6	7.98	>5000	Fear Of The Dark	master/20787	Title
7	2.739	>5000	Les Lunes De Saturne	master/1460896	Genre
8	9.669	28	Fear(s) Of The Dark	artist/5642539	Name
9	8.532	1407	Nachtgarm	artist/552210	Group
10	8.454	86	Exibit A	artist/2174612	Member

Table 32: Query: fear of the dark

Rank	Score	Init Rank	Item	Type and ID	From
1	8.694	4	Lay Down The Law	master/673376	Artist
2	7.459	274	C. Styles	artist/283031	Group
3	6.871	1	Too Dangerous	artist/3410164	Member
4	6.468	95	Dangerous LLC.	artist/277500	Name
5	5.275	619	Dangerous	master/362959	Title
6	8.694	5	Comin' Out To Play	master/147305	Artist
7	7.459	275	Hailey "Silence" ...	artist/903977	Group
8	6.75	80	TGZ	artist/4209482	Member
9	6.424	96	Dangerous (13)	artist/4755537	Name
10	8.694	6	Life Is Funny	master/147304	Artist

Table 33: Query: dangerous

Rank	Score	Init Rank	Item	Type and ID	From
1	5.34	8	The Dark Bob	artist/1062065	Group
2	5.153	>5000	Bob!	master/1441843	Title
3	4.864	815	If You Can't Get ...	master/1609375	Artist
4	4.246	4	Bob & Bob	artist/1062050	Name
5	3.704	4663	Virgo Snakes	artist/2037801	Member
6	5.181	>5000	Sylvie Nève	artist/2081422	Group
7	5.153	>5000	Bob	master/1114001	Title
8	4.246	3236	Bob, Bob, Bob An ...	artist/2644175	Name
9	5.181	623	Jean-Pierre Bobillot	artist/662919	Group
10	3.687	4758	Improvisatyrs	artist/1046709	Member

Table 34: Query: bob

Rank	Score	Init Rank	Item	Type and ID	From
1	5.009	21	The Stereo Cynics	artist/3137874	Member
2	4.863	31	Wounds Wide Open	master/197241	Artist
3	4.661	57	For For Dreamy	artist/5510873	Name
4	4.332	>5000	Mika Ahtiainen	artist/1626199	Group
5	4.863	32	IV	master/282493	Artist
6	4.767	42	Misser	artist/2815070	Member
7	4.599	117	For	artist/776348	Name
8	2.606	>5000	For	master/817721	Title
9	4.332	>5000	Eza Viren	artist/2712697	Group
10	4.863	33	Samsara	master/465071	Artist

Table 35: Query: for

Rank	Score	Init Rank	Item	Type and ID	From
1	3.406	723	And And And	artist/2178857	Name
2	3.207	915	They Found My Na ...	artist/412608	Member
3	2.882	>5000	And And And And ...	master/553080	Title
4	2.198	>5000	Nathan Baumgartner	artist/5343254	Group
5	2.011	>5000	The Failure	master/1081695	Artist
6	3.207	916	Soliah Shawcross	artist/487308	Member
7	3.023	1088	Eric Early, And ...	artist/2708362	Name
8	2.379	>5000	And.	master/364667	Title
9	3.181	941	ith/ist/ism	artist/413189	Member
10	2.198	>5000	Jonathan Sallas	artist/5343255	Group

Table 36: Query: and

Rank	Score	Init Rank	Item	Type and ID	From
1	5.11	178	Paul Et Paul	master/1406150	Title
2	5.011	>5000	Bill Jones (3)	artist/969367	Group
3	4.02	1530	Log Cabins	master/731740	Artist
4	3.71	4823	Paul Paul	artist/98607	Name
5	3.066	554	Leston Paul And ...	artist/3126009	Member
6	5.019	>5000	Paul	master/1047371	Title
7	5.011	>5000	Annie Laurie	artist/733552	Group
8	4.019	1533	Can't Wait / How ...	master/823968	Artist
9	3.71	4824	$Paul^2$	artist/2686991	Name
10	5.019	228	PAUL	master/1635162	Title

Table 37: Query: paul

Rank	Score	Init Rank	Item	Type and ID	From
1	5.286	>5000	FRANK	master/1315031	Title
2	5.282	>5000	Derrick Harvey	artist/1469519	Group
3	4.331	1072	Right In The Guts	master/706079	Artist
4	4.229	1145	Frank!	artist/3035439	Name
5	3.707	4304	Cadillac Heights	artist/156376	Member
6	5.286	235	Frank	master/683537	Title
7	5.24	106	Frank Morgan	artist/307693	Group
8	4.331	1073	The Devil Rides Out	master/1079413	Artist
9	4.229	1146	Frank Frank	artist/1762141	Name
10	5.286	>5000	Frank	master/825195	Title

Table 38: Query: frank

Rank	Score	Init Rank	Item	Type and ID	From
1	2.394	>5000	Death Waltz	master/1056081	Genre
2	5.04	1	The Black Family	artist/1899572	Member
3	4.672	757	Precious Souls	master/775102	Artist
4	4.62	877	Black Black Black	artist/3282946	Name
5	4.143	>5000	Michael Rose	artist/88266	Group
6	4.078	>5000	Black Black & Black	master/215125	Title
7	4.948	4	The Black Brothers	artist/1221459	Member
8	4.672	758	Oh Jah	master/1060901	Artist
9	4.456	1414	Black Black	artist/587159	Name
10	4.143	>5000	Puma Jones	artist/357957	Group

Table 39: Query: black

Rank	Score	Init Rank	Item	Type and ID	From
1	7.634	6	Black Me Out	master/719883	Artist
2	6.29	148	Against	artist/1425539	Name
3	6.249	585	James Bowman	artist/553943	Group
4	5.041	1117	Against	master/588859	Title
5	4.979	1	Against The Current	artist/4444059	Member
6	7.634	7	Stabitha Christie	master/1458804	Artist
7	6.249	586	Dustin Fridkin	artist/1280181	Group
8	6.157	187	Against Me!	artist/253797	Name
9	7.634	8	As The Eternal C ...	master/52276	Artist
10	5.041	1118	Against	master/57979	Title

Table 40: Query: against

Rank	Score	Init Rank	Item	Type and ID	From
1	2.075	583	Beethoven "Emper ...	master/1541036	Year
2	8.99	144	Decemberunderground	master/32617	Title
3	8.027	3	TOOTIMETOOTIMETO ...	master/1440438	Artist
4	7.924	115	Matthew Healy	artist/3480219	Group
5	7.372	24	Rapoon	artist/7029	Name
6	2.075	584	Clay & Vicki Ca ...	master/1544074	Year
7	8.027	1	The 1975	master/596580	Artist
8	7.924	116	Adam Hann	artist/3480220	Group
9	6.649	104	The 1975	artist/3231113	Name
10	2.075	585	Elton John Rock Hits	master/353939	Year

Table 41: Query: rapoon 1975 decemberunderground

Rank	Score	Init Rank	Item	Type and ID	From
1	19.254	24	Hammer And Anvil	master/283515	Artist
2	16.597	84	Jamie Willcox	artist/883223	Group
3	13.731	59	Pure Reason Revo ...	artist/337983	Name
4	19.254	8	The Dark Third	master/165401	Artist
5	10.988	1	Rage Against The ...	master/7939	Title
6	16.597	85	Paul Glover	artist/1134757	Group
7	9.081	124	Crucial Music	artist/2842589	Member
8	19.254	9	The Intention Craft	master/165402	Artist
9	16.597	86	Greg Jong	artist/1619232	Group
10	11.647	79	Rage Against The ...	artist/6853704	Name

Table 42: Query: pure reason revolution rage against the machine