Analysing a modified ranking algorithm for exploratory search

Master's thesis in Engineering Mathematics and Computational Science

MARKUS FÄLLMAN

Department of Mathematical Sciences
Chalmers University of Technology
Gothenburg, Sweden 2020

MASTER'S THESIS 2020:NN

Analysing a modified ranking algorithm for exploratory search

MARKUS FÄLLMAN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Analysing a modified ranking algorithm for exploratory search
MARKUS FÄLLMAN

Supervisor: Johan Jonasson, Mathematical Sciences
Examiner: Serik Sagitov, Mathematical Sciences

Master's Thesis 2020:NN
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Analysing a modified ranking algorithm for exploratory search
MARKUS FÄLLMAN
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Exploratory Search is a small emerging field within Information Retrieval, studying a type of searching called exploratory searching. This type of search is directed towards learning and investigating, and has recently started to draw attention. However, the field of Exploratory Search struggles with its methodology. A central problem is the difficulty to measure improvements due to that exploratory searching by definition lacks precise goals. New tools and ideas are therefore often evaluated with user studies. By focusing on describing *how* tools and ideas work, researchers can avoid the difficulty and contribute to the field. Such an indirect approach allows formulating measures that can be applied to ranked lists, which, in turn, allow using simulations with many benefits. This study showcases the approach.

The aim is to determine if a ranking algorithm modification influence the formation of groups in lists of ranked articles returned from an academic search engine. The data are generated by simulated searches and a Linear Mixed Model is used for the analysis. The main covariates represent how the ranking of a standard ranking algorithm is weighted together with the ranking according to two new criteria. The response variable consists of scores on how tightly connected the ranked articles are, with the importance of links decreasing with the depth, and comes from the application of a measure developed in the thesis.

The main result is that the level of interconnectedness between high ranking articles can be clearly and statistically significantly influenced by the modification, although the influence varies with the randomly generated queries. While more research is needed, this might be useful for controlling the articles interconnectedness when constructing a search engine. On a different level, the thesis shows how the indirect approach can be applied, that it enables using simulations, and it indicates that the approach can produce results interesting for exploratory searching.

Keywords: exploratory search, ranking algorithm, rank biased measure, citation expansion, linear mixed model.

Contents

List of Figures

List of Tables

1
Introduction

Searches can be classified as either look-up or exploratory, with the former having specific targets and the latter aims for investigating and learning [27]. The former has historically been the main goal in the research field Information Retrieval [2] and its precise targets make measuring success relatively easy [10, 2]. Success in exploratory searching is harder to judge. Its open-ended goals may have several correct answers [39] and exploratory searchers often struggle with expressing their information need [36]. This makes using look-up search tools for exploratory search difficult. Supporting exploratory searching directly has, therefore, started to draw more attention.

In scholarly search, this trend shows in that many academic search engines have started to help exploration through summarising search results and offering advanced filtering options, with notable implementations at Microsoft Academic[1], Dimensions[2], and Semantic Scholar[3]. The recent increase in ambitious new academic search engines is partly due to the opening up of access to metadata [24], supported by for example the Initiative for Open Citations, I4OC[4]. The improved access to data has also opened up opportunities for research.

When it comes to the research, however, the field of Exploratory Search is still undeveloped [42]. One central methodological issue relates to the complexity of exploratory searching. The existing measures do not capture how well new ideas improve exploratory searching and the lack of well-defined targets make it difficult to find new suitable methods [42]. The most common approach is to perform user studies aimed at describing the interaction between humans and the search system [42, 26], which is expensive and demanding. It is also difficult to do large enough user studies to determine how the impact of proposed changes interacts with the many different parts of a search engine. It would, therefore, be helpful to have measures that can be applied to simulations.

For the emerging field Exploratory Search, it is perhaps more important to assess modifications conveniently and for free, than to capture the impact on the essence of exploratory searching. Suitable alternatives are measures that only describe aspects of exploratory searching. Since these are easier to formulate, they can be applied directly to ranked lists. This opens for using simulations instead of user studies, which has many benefits.

By directing focus toward how proposed modifications change the search engine, compared to if the changes made it better or worse, the researcher can establish how the modifications can be used. Such knowledge can then become part of an array of tools that

[1]Microsoft Academic: academic.microsoft.com/home
[2]Dimensions:app.dimensions.ai/discover/publication
[3]Semantic Scholar: www.semanticscholar.org/
[4]I4OC: https://i4oc.orga

can be used when aiming more directly at exploratory searching.

To summarise, studying *how* relevant modifications affect the search engine might be fruitful since it can enable using simulations. Even though it does not try to capture the impact on exploratory searching, it can contribute indirectly. This study tries to exemplify this view and is a proof of concept of the approach.

1.1 Aim

This thesis analyses how changes to an academic search engine's standard ranking algorithm affects how the citation links between the articles in the ranked lists produced by the search engine form groups. More specifically, the aim is to determine if the ranked articles' interconnectedness can be controlled by three variables weighting together the default ranking with the ranking according to two new criteria.

1.2 Societal, ethical, and ecological concerns

There is no substantial societal, ethical, and ecological concerns with regards to this thesis. The thesis aims to contribute to navigating scientific domains. As such it could help research no matter direction and therefore contribute to unethical research as well, but given that science, in general, is still warranted, so is helping science in general. When it comes to the implementation of the project, there is no relevant harmful effect, since the project has been based around simulations implemented on a personal laptop. As such, the projects climate footprint is not much larger than the use of a single laptop, which is acceptably small for a master thesis project.

1.3 Structure

The rest of this thesis is divided into six sections: background on search engines, method, theory, analysis, discussion, and conclusion.

The method section describes how the data used in the analysis is generated through simulations and motivates choice made during the project.

The theory section describes the modification of the ranking algorithm, the measure used to score how interconnected the articles are, and introduces the Linear Mixed Model which is used to analyse the data generated from the simulations.

The analysis section consists of an investigative analysis of the data, which overviews and discusses trends and relationships, and an analysis using the Linear Mixed Model. The latter includes formulating the relevant Linear Mixed Model with its respective dependent and independent variables, checking the model's assumptions, and drawing conclusions about the data using the model.

2

Background on search engines

This sections gives an introduction to search engines and to citation expansion in the context of searching.

2.1 Introduction to Information Retrieval

Information Retrieval is a large domain centred around searching, with relevance judgement, performance evaluation, and user information needs as the larger topics [9]. According to Robertson [32], the field started in the 1960s, with Salton et al as a central group. Through the 1980s, many groups, such as Croft's and Robertson's, created the original versions of most of the retrieval models used today.

The commercial side started in the 1990s. Towards 2000, Google took over as the dominating search engine, due to its interface, crawl, and speed. The PageRank algorithm was part of the success, although it was not as important as usually thought [32, 9]. Nowadays, commercial web search engines incorporate hundreds of features in their ranking algorithms that are based around huge amounts of user interaction data [9].

2.2 Introduction to search engines

Search engines are systems for finding information in a collection, which consist of documents such as articles or websites [9]. There are three main parts to a search engine: the index, the query processing, and the user interface. The index is the database in which the search algorithm searches. It consists mainly of a vocabulary index and inverted indexes. The former contains all terms in the entire collection of documents while the latter maps each term to the documents that include it. When the index is set-up, the included documents are often processed in various ways to improve and speed up the later searches.

The query process is often centred around a ranking algorithm based on a formal retrieval model [9]. Besides the ranking algorithm, the query process usually includes several other steps, such as processing the query by removing too common words (stop words), formatting the words (stemming), and adding synonyms (smoothing). The ranking often includes how relevant the documents are for the query as well as a static ranking independent of the query.

The last of the main parts is the interface which enables the user interaction [9].

2.3 Introduction to retrieval models

Retrieval models are formal models for assigning each document a relevance score for a given query [9]. There are many popular ranking models, such as vector space models, probabilistic models, language models, and learning to rank models [22]. There is no clear best model. Croft et al assert that the language models are the dominant paradigm

[9], while the vector space models are central in implementation [38, 5, 16] and the open-source search provider Elasticsearch considers the probabilistic BM25 model as leading [16]. On top of this, machine learning, such as learning to rank models, has become successful due to the huge amount of data that web search engines have access to [32, 9], although the data outside web search is often too limited for this approach [9].

The most used model is the vector space model [5]. It was originally mentioned in the 1970s by Salton et al [9, 38], although there is a confusion about where [13]. In essence, the model represents each term present in the entire collection of documents with a dimension in a vector space. In this space, both documents and queries are then represented as vectors. For a given query, each document is then assigned a similarity score and ranked accordingly [35, 38, 9]. The currently best similarity measure is the cosine correlation [9, 38], which scores according to the size of the angle between the query vector and the document vector [9]. Aside from the similarity measure, the scoring also depends on how documents and queries are represented by vectors in the vector space. The representation is done through assigning each document or query a weight along each dimension. The weights are based on how the terms in the collection are present in the document or query. The most common weight function is the term frequency-inverse document frequency (tf-idf), which is often combined with a document length normalization [38, 9].

Another group of models is language models, which represent documents with probability distributions for all words. An example of a language model is the relevance model. Croft et al explains that it fits probability distributions over the number of occurrences of the words present in the document or query using maximum likelihood estimation for queries and each document and then compares the distributions using the Kullback-Leibler divergence measure [9].

The last retrieval model introduced here is the BM25. Since the BM25 retrieval model is used in the implementation it is here described a little more thoroughly than the other retrieval models.

The BM25 ranking algorithm is a Bayes classifier [9] developed by Spärck Jones, Walker, and Robertson in 2000 [37]. It uses a probabilistic framework, where the relevance of a document is formulated as the probability of relevance given the document specification. By using Bayes' Theorem and some independence assumptions, the relevance of the document can be formulated as a sum of how probable the content of each document field is given a relevance specification. Here, document field refers to different parts of a document, such as the title or the abstract. These probabilities are then estimated using weight functions. The weight functions are designed to score how important matches are between the relevance specification, such as a term in the query, and the document specification, such as the number of occurrences of the query term in the document.[37] The weight functions are, however, not specified by the framework. The commonly used versions are similar to both the weights used in the vector space model and to the maximum likelihood estimations used in the language models, which lessens the difference between the three retrieval models. In this thesis, the implementation of BM25 in the open-source search engine Elasticsearch is used. In Elasticsearch' imple-

mentation[30, 29], the weight function for a single field (such as the title) is

$$\sum_{i=1}^{|Q|} ln\left(1+\frac{N-n_i+0.5}{n_i+0.5}\right)\frac{f_i(k+1)}{f_i+k(1-b+b\frac{l_d}{l_{avg}})},$$

where $|Q|$ is the number of terms in the query, N is the number of documents, n_i is the number of documents with the ith query term in the field, l_d is the length of the field, l_{avg} is the field's average length for all documents with that field, and f_i is the frequency of the ith query term in the field. The variables b and k are parameters, controlling the length normalization and how much a single query term can affect the score.

2.4 Introduction to open-source search engines

There are many open-source search engines, such as Lemur Toolkit & Indri Search Engine[1], Apache Solr[2], and Elasticsearch[3] (known as Elastic). The two latter are both based on Apache Lucene search library[4]. In this work, the implementation is based around Elastic.

2.5 Introduction to academic search engines

There are a number of fully developed search engines for scholarly search, from the most used academic search engine Google Scholar[5] [24] and its web search engine twin Google[6], to recent competitors such as Dimensions[7], Semantic Scholar,[8] and Microsoft Academic[9], to open source attempts such as CrossRef[10] and CiteSeerX[11]. There is also the search systems provided by academic publishers such as Web of Science[12], Science Direct[13], Springer Nature[14], and Research Gate[15], as well as the numerous search engines offered by universities, often through the search provider Ebesco[16].

There is little information available on how most of these search engines work. When it comes to Google Scholar, Beel et al analyzed their ranking algorithm and came to the conclusion that the words in the title have a dominating influence together with citation counts, and that exact matches are important [4].

The academic search engines CrossRef, CiteSeerX, and Semantic Scholar all disclose that they use standard open source solutions together with static ranking and some

[1] http://www.lemurproject.org/
[2] https://lucene.apache.org/solr/
[3] https://www.elastic.co/
[4] https://lucene.apache.org/
[5] Google Scholar: scholar.google.com/
[6] Google: www.google.com
[7] Dimensions:app.dimensions.ai/discover/publication
[8] Semantic Scholar: www.semanticscholar.org/
[9] Microsoft Academic: academic.microsoft.com/home
[10] CrossRef: https://www.crossref.org/
[11] CiteSeerX: https://citeseerx.ist.psu.edu/index
[12] Web of Science: https://login.webofknowledge.com
[13] Science Direct: https://www.sciencedirect.com/
[14] Springer Nature: https://www.springernature.com/gp
[15] Research Gate: https://www.researchgate.net/search?q=
[16] Ebesco: https://www.ebsco.com/academic-libraries

additional features. CrossRef implements a version of the vector space model in the Apache Solr search engine (although CrossRef is transitioning to Elasticsearch according to a developer at CrossRef in a personal mail conversation). CiteSeerX, which is built upon CiteSeer, also uses Apache Solr [44]. Semantic Scholar uses a vector space model implemented in Elastic Search, combined with document features in a learning to rank architecture as well as with a static ranking based on the number of citations, recent citations, and the time of publication [45, 1].

Researchers' interest in academic search is increasing [5]. When it comes to the ranking techniques used in research projects, the most common is the vector space model with the cosine similarity [24, 5]. Apart from the vector space model, many also use BM25, which is considered best for scholarly retrieval [24]. Static rankings of papers are also important with using citation-based metrics as the most common approach [31, 40].

2.6 Introduction to citation expansion

In a search context, citation expansion means assigning relevance to articles based on their citation links. Until recently, research on citation expansions has been hindered by the limited access to citation databases [8], but research still suggests that it improves academic search [24, 25] and the interest in citation expansion is growing [19, 6].

The main argument for the usefulness of citation expansion is that citations, and especially multiple citations, are a judgement of relevance by researchers. Supporting this, Belter finds that documents with multiple citation links to relevant papers often are relevant as well [6]. Another argument for citation expansion is that it can up for vocabulary mismatch, which Croft explains happens when relevant documents do not match a query because of different wordings [9].

Citation expansion can follow citation links both forward and backward, and there are three types of relations: direct citation, bibliographic coupling, and co-citation [19, 25]. Bibliographic coupling means that two documents are referring to a third document [19] and co-citation that a pair of documents are cited by the same third document [19, 24, 20]. In research, many studies base the expansion around seed articles [23, 19, 20, 6], or expands a co-citation network by using words extracted from cited documents [20].

3
Method

This section clarifies how the different parts of this study tie together, what is done in the implementation, and how the parts are related to the aim of the thesis. First is an overview of the steps leading from the article database to the data sets discussed in the analysis. After the overview, different choices in the implementation and the choice of the statistical model are motivated. Lastly, the database and the search engine set-up are described.

3.1 Overview

As described at the beginning of the thesis, the aim is to exemplify how modifications to a search engine can be analysed in the context of exploratory searching. Specifically, this study examines modifications of a common ranking algorithm and the analysis applies a Linear Mixed Model to data gathered from search simulations.

The simulations consist of four larger steps: generating random queries, ranking with the Elasticsearch search engine, reranking according to the modifications, and scoring the result with the measure developed in this thesis. Figure 3.1 provides an overview and shows the parameters used in each step.

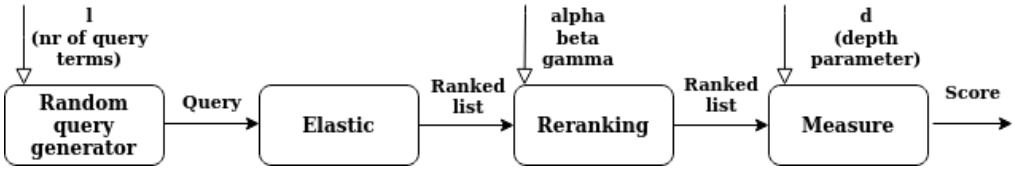

Figure 3.1 – *Overview of the steps in the simulation from the query generation to score. The vertical arrows represents parameters used in the respective box.*

In more detail, the simulations are based around the open-source search engine Elasticsearch (Elastic). The implementation of Elastic indexes a snapshot of the article metadata used in the academic search engine Semantic Search, consisting of more than 200 million documents [1]. The random query generation is based around a feature in Elastic that returns l number of random documents, with l being a parameter set in the simulation. From each of the documents' titles, a single term is selected and added to the query. The query is then sent to Elastic, which ranks according to their version of the algorithm BM25 and returns a ranked list. The documents in that list have a relevance score which the ranking algorithm modification developed in this study uses to rerank the documents. Depending on the value of three parameters, α, β, and γ, the modification reranks according to either the original BM25 score, the total BM25 score in each document's vicinity, the average score in each document's vicinity, or a combination of

the three. Here the vicinity of a document means the document itself, all its referenced documents, and all documents citing it. The weight for the original BM25 score is set by α, the weight for the vicinity total by β, and the weight for the vicinity average by γ. Since querying Elastic is a major bottleneck in the simulations, each combination of the specified values for the ranking algorithm parameters are tried for each query. In total, the ranked lists are affected by the randomly generated queries and four parameters. These parameters are the number of terms in the query, l, and the three weights given to the different ranking criteria, α, β, and γ.

After the ranked lists have been produced, the Rank Biased Cluster-measure, developed in Section 4.2.2, is used to score how tightly connected the articles are in the ranked list. The point with this measure is to capture an aspect useful for designing search engines for exploratory academic searching while avoiding the difficulty of measuring exploratory search success directly. Since a user start at the top of the ranked lists and stop at some point before the lists' end, it matters more if there is a connection early on compared to further down. The influence of connections is therefore decreased with increasing depth, with the geometric distribution controlling the decline and with a parameter d controlling the distribution. This makes the measure rank biased. Similar to how all combinations of the chosen values of the three ranking algorithm parameters are applied to each query to speed up the simulations, the measure is applied once for every value of d for each combination of query, α, β, and γ. The score resulting from applying the measure is confined to the interval [0,1]. A score at exactly zero means that all articles form separate clusters and a score at exactly one means that all articles are connected with at least one article higher up in the list, creating a single cluster of articles. In practice, however, both scores at zero and one can be reached by rounding down or up, respectively. This is because connections found far from the top of the list have little influence on the total score.

In summary, in each simulation, the ranking algorithm modification developed produces a ranked list. The list is then scored according to how closely connected the articles are using the Rank Biased Cluster-measure.

After the ranked lists have been scored, the data for the analysis is collected. For each simulation, one value is stored for each variable in the analysis data set. There is one dependent variable, the score, and six independent variables. The labelling of the variables as dependent and independent comes from the Linear Model framework used later in the thesis. The custom refers to that the values of the dependent variable are assumed to depend on the variables used to predict it. Returning to the variables, the independent variables correspond, in turn, to the parameter controlling the query length, the ID for the random query, to the three parameters used to modify the ranking, and the parameter controlling the measure. How the variables relate to the simulations are presented in Figure 3.2.

Besides these variables, there are numerous other candidate variables, since there are many different ways set-up in Elastic. A few examples of what can be done is to change the ranking algorithm used in Elastic, add a static ranking, remove stop words differently, and add a stemmer. Such changes would likely affect the resulting score and also likely interact with the ranking algorithm modifications. In a larger study, it would be important to examine many such interactions to perform what Ferro and Silvello call a component-wise analysis [21]. Here, however, the time constraints on the thesis limit the study to the variables already mentioned above.

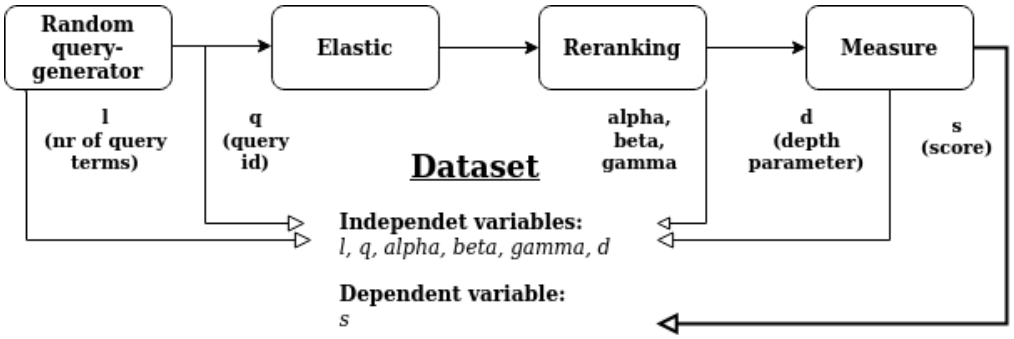

Figure 3.2 – *Overview of how the data sets' variables are collected from the simulations. For each simulation, a value for each variable is stored. The vertical arrows shows where the values of the variables are taken from.*

3.2 Choosing size of citation expansion

The ranking algorithm only considers each documents' references and citations. It is, however, easy on a theoretical level to consider the score of documents further away. The reason why only direct references and citations are used is that the networks for academic documents often grow fast, making the implementation difficult when documents further away are included.

3.3 Choosing parameter values for the simulations

In the data sets, there are five covariates that correspond to parameters in the simulations. These are l, α, β, γ, and d. Below, each parameter used in the simulation is discussed briefly and the chosen parameters values stated and argued for. These values then become the values of the covariates in the data sets.

The main reason for not trying a larger range of values is that the number of observations grows quickly when each combination of the relevant four parameters is tried for each query (the number of terms in the query, l, is unique for each query, so it does not add to the number of observations).

The parameter l controls the number of terms in the query. In the simulations, it takes the values $\{3,5,7\}$. The values were chosen to represent typical query lengths and although there probably are many queries with only two words and also some queries longer than seven, these lengths are fairly typical.

The values of the modification parameters α, β, and γ was first chosen as $\{0, 0.25, 0.5, 0.75, 1\}$, aiming for an even spread over the interval [0,1]. While it is possible to choose values outside [0,1], values larger than one are uninteresting and negative values would make the interpretation messy. The reason why large values are uninteresting is that what matters is the relation between the three parameters, which means that the effect of increasing the value of one of the parameters can already be achieved by decreasing the others. After some preliminary analysis, the values $\{0, 0.25, 0.5, 0.75, 1\}$ were complemented with the values $\{0.9, 0.95\}$ for α and with $\{0.1, 0.05\}$ for β and γ in order to follow up on an interesting pattern in the data. For the final analysis with the Linear Mixed Model, the original values in $\{0, 0.25, 0.5, 0.75, 1\}$ are used.

The parameter d controls how fast the importance of entries the ranked lists decreases with the depth. The parameter takes the values $\{0.8, 0.9, 0.95\}$. The value $d = 0.8$

was included since it is used by Ferro and Silvello in their study [21] and roughly corresponds to that the 15 first entries in the ranked lists determine 95% percent of the final score. For the two other values, 0.9 and 0.95, the corresponding sizes are roughly the 30 and 50 first entries, respectively. These values were judged reasonable since the number of entries needed for setting the 95% of the score are relatively evenly spread out.

3.4 Choosing a statistical model

This part of the method section explains why the Linear Mixed Model is a good choice for basing the analysis on and comments on what the alternatives are. First comes a motivation for selecting a linear model and a short introduction to different linear models, then an explanation of two aspects of the simulations important for selecting the right type of linear model, and finally arguments for selecting the Linear Mixed Model from the remaining relevant options.

3.4.1 Linear Models

The two most common ways to quantify the analysis in all of science is through using the frameworks regression and analysis of variance (ANOVA) [33], while ANOVA and different t-tests are dominant in the Information Retrieval field [34]. These frameworks can be generalised to the regression model General Linear Model [33]. The familiarity and range of the General Linear Model and its generalisations, therefore, makes it a fitting choice for a discussion of a more proper statistical approach in a subfield to Information Retrieval.

Linear Regression models are, however, a broad class of models. Common to all is that the mean of a dependent variable is modelled as a linear combination of one or more independent variables [7]. There are many related, more or less general, versions of linear models and, unfortunately, some confusion about how to name the versions. For clarification, some of the relevant models are described below, using the names the way they are used in Wikipedia's series on Regression analysis.

The simplest of the linear regression model is called Simple Regression, which predicts a single normally distributed dependent variable from an independent variable. The closest larger model is the Multiple Linear Regression, which can have any number of independent variables. The next, more versatile model, is the General Linear Model, which allows for predicting many dependent variables and for using both continuous and discrete variables.

From the General Linear Model, there are two different types of abstractions relevant for this study. The first is the Generalized Linear Models, which allows for assuming other distributions for the predictor variables than the normal distribution. The other type of model is the Linear Mixed Model, which is developed to handle data that is grouped into clusters centred around a random factor. The two type models, Generalized Linear Models and Linear Mixed Model, can also be combined into the Generalized Linear Mixed Model.

In this study, the Linear Mixed Model is used. This is also the model recommended by Ferro and Silvello, when showing how to take into account the interaction between the proposed changes and other parts of the search engine [21].

3.4.2 Repeated measures and mixed effects

There are two particularly relevant aspects of the simulations for choosing the Linear Mixed Model as the statistical model. The first is that the simulations use repeated measurements, and the second is the studied effects are a mix of fixed and random effects.

Starting with the explanation of the first aspect, repeated measures are defined by that the dependent variable (the simulation scores) are grouped by a source of randomness (the random queries). Since the same query is tried with all ranking algorithm modifications, the scores for that query are statistically dependent.[33] The reason for using repeated measures in the simulations is that the size of the database makes querying Elastic the by far largest bottleneck. It is, therefore, important to send queries to Elastic as seldom as possible. However, according to Rutherford, there is also a benefit to using the repeated measurements in that the variance due to the random queries is likely to be less important [33].

The second aspect is that the simulations create a mix of fixed and random effects. When the independent variables are analysed as fixed-effects it is assumed that all their relevant parts are represented [33]. A simplified example is if a variable represents the age of a person in a study of length, both children and teenagers need to be included, but it is not as important to include adults (since it is known that adults no longer grows after a certain age). In the analysis data sets, all variables that correspond to parameters in the simulations are studied as fixed. The reason is that their values, although few, are chosen to be representative. In contrast, independent variables analysed as random-effects are assumed to be only a random sample of a population of values [33]. A typical example of the latter is using human subjects. It is seldom reasonable to handpick people for a study since it would be difficult to be sure that they are representative and participants are therefore selected through random sampling. In the simulations, the queries are randomly sampled from a vast number of possible queries and it is hard to know how representative they are. When analysing the data sets, the variable representing the random queries, q, thus needs to be studied as a random effect.

In total the analysis data sets consists of variables representing mixed-effects created from repeated measurements. There are clusters around random factors in the data, which is what Mixed Models are designed to handle.

3.4.3 Linear Mixed Models

The reasons for choosing the Linear Mixed Model instead of other Mixed Models are discussed here. Besides the Linear Mixed Model, the two main alternatives are the Logistic Regression, i.e. a Generalized Linear Mixed Model with the binomial distribution, and the Beta Regression Model.

The argument for the Linear Mixed Model is that it is the simplest Mixed Model, making it the default choice. There is a problem, though, in that the dependent variable here only takes values in the interval $[0, 1]$ which do not fit the normal distribution assumed in the Linear Mixed Model. Breaking the normality assumption this way is, however, common when using measures in the Information Retrieval field [7].

The closest alternative is to use the Generalized Linear Mixed Model with the binomial distribution, which does give predictions in the right interval. The problem with the binomial distributions is that it is intended to predict the probability that the dependent variable equals one instead of predicting the dependent variables value directly. It can be

used, but using the model becomes more difficult.

The third relevant model is the Beta Regression Model. The argument for it is that the data fulfils all the model assumptions. The Beta Regression Model is also very flexible. The problem with it is that it is rather more involved and the theory behind is not as well developed.[12]

Given the time constraints for this thesis, the most reasonable of the three alternatives is the Linear Mixed Model. Although the data do not fit its assumptions completely it can still be used. This is supported by Carterette, who, when examining the influence of breaking the distributional assumption in Information Retrieval experiments, concludes that the violation only has an acceptably small effect on the resulting model [7]. Another reason against the two other alternatives is that they are much more technical than the typical analysis used in Information Retrieval. This makes them unnecessarily complicated compared to the Linear Mixed Model.

3.5 Implementation

This part describes the implementation in more detail.

3.5.1 Article database

The article data that is indexed with the Elastic search engine is a snapshot of the article metadata from the Semantic Scholar academic search engine which has been made available for research. The data is thoroughly described in Ammar et al's article *Construction of the Literature Graph in Semantic Scholar* [1]. In summary, it consists of the metadata of close to 200 million documents (the actual number of documents is 177 443 710), each with many fields such as title, abstracts, incitations, and outcitations. For this study, only the fields title, incitations, and outcitations were kept from the metadata to increase the speed of the Elastic search implementation. The reason why the abstract was not included was that the size of the database made it difficult to handle on the laptop used for the simulations.

The data set could have been trimmed further by e.g. removing all articles but those written in English. However, using a large set of metadata in the implementation is important since when the citation expansion modifies the ranking, the documents that are linked to must also be in the metadata set if the expansion is to work properly. The full data set was therefore kept.

3.5.2 Elasticsearch

The search engine used in the experiments is the open-source search engine Elasticsearch (known as Elastic) [14]. Elastic is built upon the Apache Lucene search library. [15]. It was used with the default settings. Elastic was chosen both since it is relatively easy to operate and since it is a popular choice.

When setting up the index, the data was put in without any relevant pre-processing, except the already mentioned discarding of other fields than ID, title, citations, and references. It was not necessary to remove stop words (common words like 'and') to improve the precision of queries since the retrieval model BM25 assigns them very little weight. However, doing so might have increased the speed of the database and also have improved the generation of the randomised queries. In hindsight, it would also have been useful to apply a stemmer that would have reduced words to common forms.

When it comes to the ranking algorithm, the ranking algorithm BM25 was used with the weight function as described in Section 2.3. The two parameters in the weight function are left at their default value. There are several other ranking functions available in Elastic, such as the commonly used vector space model [18], and even some more advanced smoothing algorithms [17]. Although it would have been interesting to try different ranking algorithms, doing so requires reindexing the database and it was, therefore, not done due to the project's time constraints.

The set-up differs from a complete search engine in several ways. Online academic search engines naturally also index the abstracts and the body of the articles, although this difference should not affect the results as much as one might believe since most weight is usually given to the title. Another difference is that the implementation in the thesis does not use any advanced static ranking (such as Page Rank) to boost quality articles, although a feature is used that punishes articles without either references or citations. Another similar static ranking that is not added is the recency boost that can be given newer articles to compensate for that older articles have accumulated more citations. Furthermore, it was also out of the scope of the thesis to use more advanced additional information, such as author prominence, upcoming events, trends, and so forth.

4

Theory

This section has three large parts: a description of the ranking algorithm modifications, a description of the Rank Biased Cluster-Measure, and a description of the Linear Mixed Model framework.

4.1 The ranking algorithm modification

The ranking algorithm modification is described in the following section. Even if the context is exploratory academic searching, the modification is discussed in a more general wording, by for example referring to documents as nodes. The idea is to allow it to be applied to other areas than articles with citation links, such as the more general area knowledge graphs. Along this line, the modification is formulated independently of which retrieval models that it is used together with and the retrieval model used is referred only to as an independent scoring function.

Moving on to the mathematical description, assume that there is an undirected graph with nodes, with D_i being the ith node. Further assume that there is some arbitrary scoring function $S_O(\cdot)$, with O for original, which can be used to score each node. In the context of this study, the nodes are documents, the scoring is done through the BM25 algorithm, and the links consist of references and citations. Given a natural number, r, define a r step-vicinity, $V_r(D)$, around a node, D, as the set consisting of D together with all nodes up to r steps away in the graph. In the simulations, the vicinity $V_1(D)$ is used and it includes the document D itself together with its referenced documents and the documents that cite it.

Define a vicinity scoring function, $S_V(D, S_O, r, \beta_r, \gamma_r) \in (-\infty, \infty)$, which calculates the score of a node D, based on the score given by the original scoring function, S_O, to the nodes in the r step-vicinity around the node. The function uses two parameters, $\beta_r \in (-\infty, \infty)$ and $\gamma_r \in (-\infty, \infty)$, to combine the total vicinity S_O score and the average vicinity score. If the values for β_r are restricted to $[0, 1]$ and γ_r is set to $1 - \beta_r$, the combination becomes an interpolation between the total and the average vicinity score, but there is no need to commit to such a restriction. In other words, $S_V(D, S_O, r, \beta_r, \gamma_r)$ is defined as

$$S_V(D, S_O, r, \beta_r, \gamma_r) = \beta_r \sum_{j:D_j \in V_r(D)} S_O(D_j) + \gamma_r \sum_{j:D_j \in V_r(D)} \frac{S_O(D_j)}{|V_r(D)|}.$$

The vicinity scoring function can now be used to create a new scoring function, S_M, with M for modification, that combine different sized vicinities. As discussed in Section 3.2, S_M is here limited to combining $S_V(D, S_O, 0, \beta_0, \gamma_0)$ and $S_V(D, S_O, 1, \beta_1, \gamma_1)$, and only this case is described. Let $\omega \in (-\infty, \infty)$ and $\eta \in (-\infty, \infty)$ be two parameters that

weight the two vicinity functions. Define $S_M(D, S_O, \omega, \eta, \beta_0, \gamma_0, \beta_1, \gamma_1)$ as follows

$$
\begin{aligned}
&S_M(D, S_O, \omega, \eta, \beta_0, \gamma_0, \beta_1, \gamma_1) = \omega S_V(D, S_O, 0, \beta_0, \gamma_0) + \eta S_V(D, S_O, 1, \beta_1, \gamma_1) = \\
&= \omega\Big(\beta_0 \sum_{j:D_j \in V_0(D)} S_O(D_j) + \gamma_0 \sum_{j:D_j \in V_0(D)} \frac{S_O(D_j)}{|V_{0,S_O}(D)|}\Big) + \\
&+ \eta\Big(\beta_1 \sum_{j:D_j \in V_1(D)} S_O(D_j) + \gamma_1 \sum_{j:D_j \in V_1(D)} \frac{S_O(D_j)}{|V_1(D)|}\Big).
\end{aligned}
$$

Next, the formula can be simplified since $|V_0(D)|$ equals one. This leads to that

$$
\begin{aligned}
&S_M(D, S_O, \omega, \eta, \beta_0, \gamma_0, \beta_1, \gamma_1) = \omega S_V(D, S_O, 0, \beta_0, \gamma_0) + \eta S_V(D, S_O, 1, \beta_1, \gamma_1) = \\
&= \Bigg[\omega\Big(\beta_0 \sum_{j:D_j \in V_0(D)} S_O(D_j) + \gamma_0 \sum_{j:D_j \in V_0(D)} \frac{S_O(D_j)}{|V_{0,S_O}(D)|}\Big) = \omega(\beta_0 + \gamma_0) S_O(D) = \\
&= [\omega(\beta_0 + \gamma_0) = \alpha] = \alpha S_O(D)\Bigg] = \alpha S_O(D) + \eta S_V(D, S_O, 1, \beta_1, \gamma_1) = \\
&= \alpha S_O(D) + \eta\beta_1 \sum_{j:D_j \in V_1(D)} S_O(D_j) + \eta\gamma_1 \sum_{j:D_j \in V_1(D)} \frac{S_O(D_j)}{|V_1(D)|} = [\eta\beta_1 = \beta, \eta\gamma_1 = \gamma] = \\
&= \alpha S_O(D) + \beta \sum_{j:D_j \in V_1(D)} S_O(D_j) + \gamma \sum_{j:D_j \in V_1(D)} \frac{S_O(D_j)}{|V_1(D)|} = S_M(D, S_O, \alpha, \beta, \gamma),
\end{aligned}
$$

with $\alpha \in (-\infty, \infty)$, $\beta \in (-\infty, \infty)$, and $\gamma \in (-\infty, \infty)$. As with $S_V(D, S_O, r, \beta_j, \gamma_j)$, the parameters α, β, and γ can be chosen so that the combination becomes an interpolation between the original retrieval model matching score given by $S_O(\cdot)$ and between the vicinity's total and average scores. Furthermore, even though the parameters can formally take any real values, it is only interesting to define them in relation to each other. This is because the ranking of the nodes does not change if the scores are multiplied by some constant. In the simulations, the values for the parameters are chosen within $[0, 1]$.

4.2 The Rank Biased Cluster-measure

This section describes the Rank Biased Cluster-measure that is used to score how closely interconnected the ranked lists produced in the simulations are. Although this measure will be used to score the documents ranked according to their score given by the ranking algorithm modification described in the previous section, this measure is separate from how the ranked lists were ranked and from the score that the lists are ranked according to. Instead, it only uses the articles' citation links and position in the list. It can, therefore, be used with any ranking algorithm.

4.2.1 Rank Biased measures

Rank Biased measures are a type of measure developed by Webber, Moffat, and Zobel for their Rank Biased Overlap-measure [41]. This type of measures give more weight to the beginning of ranked lists, since a user starts to examine ranked list at the top and stop at some depth, usually without having seen the entire list.

The tool used by Webber, Moffat, and Zobel to model the user's behaviour when formulating their Rank Biased Overlap-measure, is to bias a scoring at each depth by a decreasing probability. This leads to that the impact of the beginning is not overshadowed by the rest of the list. The probabilistic interpretation is that the user starts at the top and then have a given probability to stop before each next entry. The aspect of the list that the measure tries to capture is judged (i.e. scored) at the depth where the user stops. Since this depth is random, the Rank Biased measures score the list according to the expected value of user scores.[41] This idea will be used to develop the Rank Biased Cluster-measure below. First, though, is a description of the general form of Rank Biased measures.

Building on Webber, Moffat, and Zobel's approach, consider the family of measures of the form:

$$M(A,w) = \sum_{k=1}^{\infty} w_k A_k$$

where w is a vector of weights capturing the distribution of where the user stops and A is a vector of scores between zero and one for each depth. How the vector of scores is defined depends on which Rank Biased measure it is. In the Rank Biased Cluster measure below, A_k will represent the interconnectedness between the k highest ranked articles. Then $0 \leq M \leq \sum_{k=1}^{\infty} w_k = 1$, and each A_k has a fixed contribution w_k.[41]

For the weights, Webber, Moffat, and Zobel use the geometric distribution [41], which will also be used here. A difference in how they define the distribution is that the parameter normally used, $p \in (0,1]$, is defined as $1-p$. This thesis continues their usage. Another difference in the description below compared to the standard usage of the geometric distribution is that the parameter p is called d. This is to avoid confusion in the analysis section when discussing both the distribution parameter and p-values, i.e. the probability of getting at least as extreme values given a tested hypothesis. To sum up, the parameter used here is $d \in [0,1)$.

Following Webber, Moffat, and Zobel [41], let the kth weight have the value $(1-d)d^{k-1}$, for $d \in [0,1)$. The rank-biased measure then becomes

$$M_{RB}(A,d) = (1-d)\sum_{k=1}^{\infty} d^{k-1} A_k.$$

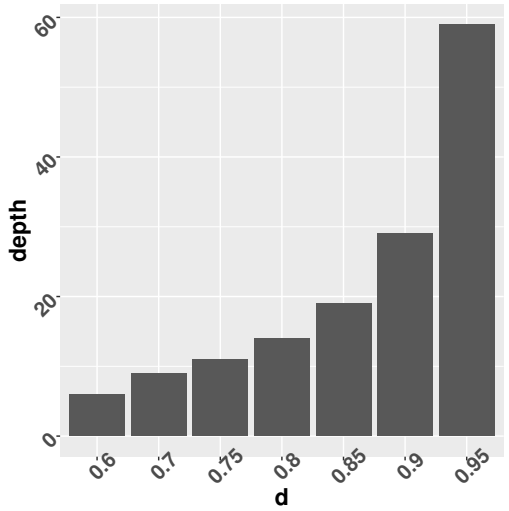

Figure 4.1 – *Depth needed in rank biased measures to reach 95% of the maximal score for different values of d. The calculation assumes an infinite lists with a maximal score for each rank.*

In the formula, d decides how fast the weights decrease, with a smaller d leading to a larger focus on the start of the list. By choosing d close to one, the important part of the infinite ranked list can become arbitrary long. Real lists produced by search engines do naturally not have an infinite length. Despite this, it is a reasonable approximation, since most search result lists in practical cases are long enough for the weights to become insignificantly small (for reasonable choices of d). Furthermore, the application of the measures will in a practical application often need to use a cut-off point to speed up the computations. This then gives a lower bound on the score. Where such cut-off points can be set depends on both the precision needed and on the used parameter d. The latter dependence is illustrated in Figure 4.1, where the depth needed to reach 95% of the max score (given a maximal score at each rank) is shown for different values of d.

4.2.2 The Rank Biased Cluster-measure

This measure attempts to capture how connected a single ordered list of results is while giving more importance to connections between the top items. It will be referred to as the Rank Biased Cluster-measure.

Let C_k be the number of clusters in the list at depth k, with k starting at two and with a cluster defined as a group of documents in the list above or at depth k in which all documents can be reached from any of other using an arbitrary number of intermediate steps within the group.

Then define the score at depth k, A_k, as

$$A_k = \frac{(k-1)-(C_k-1)}{k-1} = \frac{k-C_k}{k-1},$$

which fulfils the requirement that $A_k \in [0,1]$. The subtraction of one from each term is needed since there is always at least one cluster, while, on the other end, can be at most k clusters. The minuses therefore allow A_k to take the values zero and one.

The score $s_{rcb}(d)$ for the Rank Biased Cluster measure becomes

$$s_{rcb}(d) = (1-d)\sum_{k=2}^{\infty} d^{k-2}\frac{k-C_k}{k-1}.$$

4.3 The Linear Mixed Model

In this section, the theory behind the Linear Mixed Model is described. Since the Linear Mixed Model is an extension of Linear Regression, the General Linear Model is first described as a brief introduction to linear models.

4.3.1 Linear Models

General Linear Models describe, according to Rutherford [33], each observation of the dependent variable, y, with the formula $y = Prediction + Error$. The prediction term captures the influence of the independent variables on y. It consists of a linear combination between the independent variables together with a constant and their respective estimated influence. The constant is called intercept and adds a default value for y. The error term captures that y might not be completely determined by the intercept and the independent variables. In the General Linear Model, the error term is assumed to be normally distributed, with a zero mean and the variance estimated when fitting the model.

Following Olofsson and Andersson [28], let the n observations, Y, be an $n \times 1$ vector of random variables, the k independent variables (including the intercept) be a $n \times k$ matrix X with know values, λ be a $k \times 1$ vector, and the errors, ε, be a $n \times 1$ vector with each element independently distributed according to $\mathcal{N}(0, \sigma^2)$. The General Linear Model can then be written as

$$Y = X\lambda + \varepsilon,$$

where λ and σ^2 are estimated when fitting the model.

4.3.2 The Linear Mixed Model

The Mixed Model inherits most of the framework of the General Linear Model [11]. The difference is that the formula for the Linear Mixed Model includes a random term as well on top of the sum of the fixed prediction term and the error term used in the General Linear Model. This is due to that in the Linear Mixed Model, the data is assumed to be organised in clusters within which the observations are dependent. The clusters themselves are, however, assumed to be independent of each other. As Wiley and Wiley explain [43], there are many types of possible dependencies within clusters, each with its corresponding Linear Mixed Model. A good example is the most simple version, the random intercept-only model. It has a different mean for each cluster, but apart from that the observations behave as in the General Linear Model around their respective cluster mean.

The version of the Linear Mixed Model that will be used in this study is the random slope and intercept-model, which adds random slopes to the random intercept-only model. On top of having different means for each cluster, Wiley and Wiley explain [43], the random slopes let the influence of random factors vary between clusters. Since both the random intercept and random slopes depend on the cluster, they are often correlated and therefore described using a multivariate normal distribution.

The reason why the random slope and intercept-model is suitable for the data generated in this study is discussed in section 3.4, but in short, the scores from the measure likely varies with the different queries and the ranking algorithm modification depend on which ranked list it modifies. Thus, the random query's influence on the score creates a random intercept and the query's influence on the effect of the modifications creates a random slope.

The Linear Mixed Model has five main assumptions [43]. The first three concern the fixed term and the error term. These are that the relationship between the fixed factors and the overall mean is linear, that there is no trend (homogeneity) in the residuals, i.e. the differences between the predicted values and the actual values, and that the residuals are normally distributed. The last two assumptions concern the random effects. These are assumed to be independent between the different cluster and multivariate normally distributed within.

Following Demidenko [11], let i indicate the ith cluster with $y_i \in \mathscr{R}^{K\times 1}$ being the K observations within the ith cluster. Let the values of the m variables with a fixed contribution, including the fixed intercept, be described by the matrix $\mathbf{X}_i \in \mathscr{R}^{K\times m}$ and the vector $\lambda \in \mathscr{R}^{m\times 1}$ be the coefficients describing their influence. Let the values of the l variables with a random contribution, including a random intercept, be described by the matrix $\mathbf{Z}_i \in \mathscr{R}^{K\times l}$ and $\eta_i \in \mathscr{R}^{l\times 1}$ be the random coefficients describing their influence. Finally, let the independently and identically normally distributed errors be described by the vector $\varepsilon_i \in \mathscr{R}^{K\times 1}$. The model can then be written as

$$\begin{aligned} y_i &= X_i\lambda + Z_i\eta_i + \varepsilon_i, \\ \eta_i &\sim \mathscr{N}(0,\Sigma), \\ \varepsilon_i &\sim \mathscr{N}(0,\sigma^2 I), \end{aligned} \tag{4.1}$$

with $X_i\lambda$ capturing the fixed contribution to the observations, $Z_i\eta_i$ the random contribution, and ε_i the variance left unexplained by the model. In this formulation, it is the parameters λ, Σ, and σ^2 that are estimated when the model is adapted to the data.

4.3.3 The Likelihood Ratio Test

The Linear Mixed Model is harder to work with than regular Linear Models since it is more complex. The former is also more of an open area of research and the community much smaller, leading to that there are often situations where there are no clearly correct option. There are, for example, no standard tests comparable to the F-test (since it is not always possible to know the degrees of freedom) nor is there a default goodness of fit statistic comparable to the Linear Model's adjusted R^2 value.

Despite the difficulties, it is important to test how significant the different variables in a chosen model are. Through such testing, variables that are not central to the model can be eliminated. This is important in order to avoid over-fitting, which happens when a too advanced model finds trends in the data that are only there by chance.

In a Linear Mixed Model, there are several approaches for testing the significance of variables. The one chosen for this thesis is to compare nested models with likelihood ratio tests. A nested model is a smaller version of another model, i.e. the full model includes all variables from the smaller model and some more. When the likelihood ratio test is applied in the analysis, the difference between the nested model and the full model

will only be a single variable. This is because the test is then more accurate for Linear Mixed Models.

Likelihood ratio tests compare two proposed models about the data, quantifying if the difference between them is significant. The test estimates how likely the data is if the model is true for each model. The likelihood of the data given the smaller model is then divided with the likelihood given the larger model. The version of the test used in the analysis, provided by the anova function in R car package, computes the double negative logarithm of the ratio and compares it with a χ^2 distribution. This approach is based on Wilk's theorem.

4.4 Intraclass correlation coefficient and effective sample size

Two tools that can be used to get an understanding of clustered data are the effective sample size, $N_{effective}$ and the intraclass correlation coefficient, I_{ICC}.

The effective sample size gives an indication on how many independent samples the data corresponds to. This is useful for getting an intuition about the data, since independent samples are easier to understand. As Wiley and Wiley explains [43], the formula for $N_{effective}$ depends on the number of independent clusters, N, and the number of observations per cluster, k, and the intraclass correlation coefficient (described below). The formula is as follows

$$N_{effective} = \frac{Nk}{1+(k-1)I_{ICC}}.$$

The intraclass correlation coefficient captures how much of the variance in the data that comes from having different clusters in the data. It does so by dividing the variance between clusters with the sum of the variance between and within the clusters. According to Wiley and Wiley [43], the estimation of the variances depends on the implementation, but given the variances the formula for the intraclass correlation coefficient is

$$I_{ICC} = \frac{\sigma^2_{between}}{\sigma^2_{between} + \sigma^2_{within}}.$$

5
Analysis

The following section is divided into two parts, where the first explores the results from the simulations and the second part applies the Linear Mixed Model. The two parts use two different data sets, which will be referred to the exploratory data set and the model data set below. Both data sets are generated through the simulations. The difference between them is that the model data set has fewer variables and more observations per variable combination. The reason why a second data set was generated is that more observations per variable combination helps the model to handle the large variance in the data that is due to the randomly generated queries.

5.1 Exploring the data

This part of the results overviews and explores the relations between the variables in the data. Besides building an understanding of the relationships between the variables in the data set, it is the observations in this part that is used to select the variables that are included in the Linear Mixed Model.

5.1.1 Overview of the data set

The variables in the exploratory data set are displayed in table 5.1. The meaning of each variable is summarized here (see Section 3.1 for a more complete overview). The scores given by the Rank Biased Cluster-measure are denoted with s and form the dependent variable later in the model. The other variables are all among the model's independent variables.

The variable q represents an ID of the randomly generated queries, the variable l represents the number of words in the query, and the variable d corresponds to a parameter used in the Rank Biased Cluster-measure, with larger values of d representing that a larger part of the final ranked list can influence the score.

The three remaining variables, α, β, and γ, are the variables of main interest. Together they correspond to how the ranking score given by the search engines default BM25 ranking algorithm is combined with the total score and the average score in each document's vicinity. The combination $\alpha = 1, \beta = 0, \gamma = 0$ ranks according to the B25 algorithm, $\alpha = 0, \beta = 1, \gamma = 0$ according to the vicinity total, and $\alpha = 0, \beta = 0, \gamma = 1$ according to the vicinity average.

As can be seen in Table 5.1, the total number of scored observations (s_{rbc}), is much larger than the number of separate queries. This is because the simulations apply each combination of the main parameters, α, β, and γ, to each query and then evaluates each resulting ranked list once for each value of d. The total number of observations thus becomes $300 \times (7 \times 7 \times 7 - 1) \times 3 = 307800$. The minus one account for that the combination $\alpha = \beta = \gamma = 0$ has been removed since it gives a zero score for all documents.

Table 5.1 – *Variables in the exploratory data. The variable s is the Rank Biased Cluster score, q is an arbitrary assigned ID for each query, and l is the length of the query (the number terms). The variables α, β and γ represents the three parameters controlling the ranking algorithm modification, and the variable d represent the depth parameter used in the measure.*

Variable	Values	Total
s	$\in [0,1]$	307800
q	$\in \mathbb{N}$	300
l	{3,5,7}	
α	{0, 0.25, 0.5, 0.75, 0.9, 0.95, 1}	
β	{0, 0.05, 0.1, 0.25, 0.5, 0.75, 1}	
γ	{0, 0.05, 0.1, 0.25, 0.5, 0.75, 1}	
d	{0.8, 0.9, 0.95}	

5.1.2 Variance due to the random queries

The left panel in Figure 5.1 overviews the distribution of the scores. Particularly interesting is a large spike in the density close to the maximal score 1. Something else to notice is that the density never comes close to zero for the lower end of the scores, which will be important to consider when fitting the model. These points will be returned to further down after the influence of the different factors have been examined.

The right panel in Figure 5.1 shows the distribution of mean scores for each query. The averaging over the main variable and the depth variable does not influence the overall appearance very much. The one notable change is the disappearance of the large spike in the density for scores close to one. This means that the spike was not there for all variable combinations.

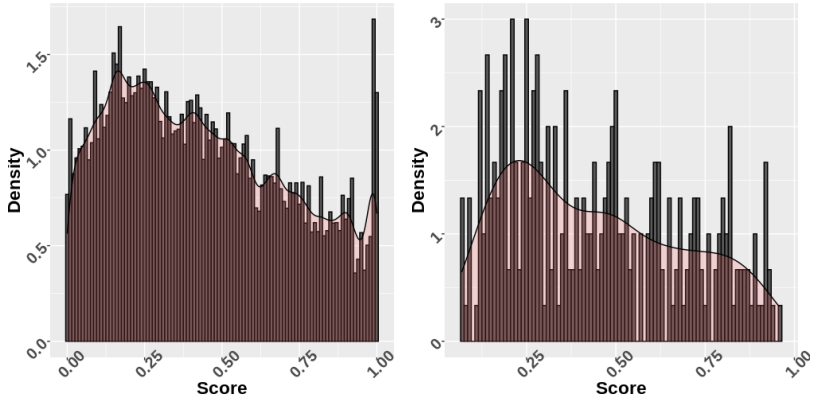

Figure 5.1 – *Overview of the distribution of scores for the exploratory data set. Left: the distribution of all scores. Right: the distribution of mean scores for each query.*

The intraclass correlation coefficient and the effective sample size for the exploratory data set are displayed in Table 5.2. The strikingly large variation between the queries, which each form an independent cluster of measurements, accounts for almost 80% of the total variation. This large variation between the queries means that a large sample is needed to investigate the main factors. This can also be seen in that the actual sample size is almost 800 times larger than the calculated effective sample size.

Table 5.2 – *The intraclass correlation coefficient and effective sample size for the exploratory data set.*

Type	ICC	Total
Queries (independent clusters)	0.76	300
Residual	0.24	
Sample Size		307800
Effective Sample Size		394

5.1.3 Main variables

In the following part, the relation between the variables α, β, and γ, and the score is explored. As previously described, these three variables correspond in turn to weights given to each document's original BM25 ranking score, its total vicinity score and its vicinity average. Further down, the analysis will focus only on an interpolation between two of these three variables while setting the third to zero. Before discussing that though, the three variables and their relationships to the score are introduced here.

To get an initial feel for that the influence of the three variables on the score depend on each other, the mean scores over the variable values are calculated separately for each variable and plotted together in the left panel of Figure 5.2. From the mostly horizontal lines in the figure, it is evident that neither of the main variables dominates the result on their own. This can be compared with a much larger change in mean score shown in the three panels to the right of the figure. Here the mean score over variable values is calculated for each of the three variables, given that one of the other main variables is zero. The reason for only setting one of the variables to zero at a time is that the modification to the ranking algorithm theoretically only changes with the relation between its parameters. Therefore, if two variables are zero, changes in the third does not change the score.

When comparing the different panels in the right part of Figure 5.2, it is clear that the relation between either α and β or α and γ is of the most interest, which is reasonable since those to relations concerns how much the original ranking is changed, while the relation between β and γ is about whether the citation expansion is normalised.

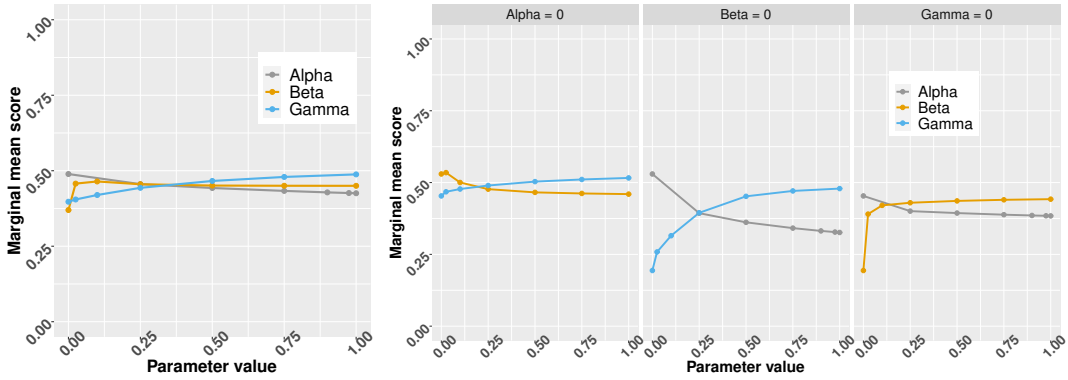

Figure 5.2 – *The mean scores over the values of the three main variables. Left: the mean for each the main variables. Right: the mean for two of the main variables, while the third is set to zero. The variables set to zero are, in turn from the left subpanel, α, β, and γ.*

To investigate the impact of the ratio between pairwise combination of the main variables, three new variables is formed by pairwise interpolations of the three main vari-

ables. The new variables are $t_{\alpha\beta} \in [0,1]$, $t_{\alpha\gamma} \in [0,1]$, and $t_{\gamma\beta} \in [0,1]$ and they fulfilled the following relations:

$$\begin{aligned} t_{\alpha\beta}: &\quad \alpha = t_{\alpha\beta}, \quad \beta = 1 - t_{\alpha\beta}, \\ t_{\alpha\gamma}: &\quad \alpha = t_{\alpha\gamma}, \quad \gamma = 1 - t_{\alpha\gamma}, \\ t_{\gamma\beta}: &\quad \gamma = t_{\gamma\beta}, \quad \beta = 1 - t_{\gamma\beta}. \end{aligned}$$

Figure 5.3 shows mean scores and boxplots over the values of the interpolations, each calculated with the respective third variable at zero. Although the information conveyed by plotting the means is also included in the boxplots, the mean scores are shown separately in the figure's left panel for a clearer view. As noted above, the trade-off between ranking according to the vicinity total and the vicinity average, which in the figure is represented by the interpolation between β and γ, only weakly affect the score. The two other trade-offs, $t_{\alpha\beta}$ and $t_{\alpha\gamma}$, both clearly show that the mean score decreases when the original BM25 score is given a larger influence. When comparing $t_{\alpha\beta}$ and $t_{\alpha\gamma}$, it is clear that while $t_{\alpha\beta}$ changes the mean score less than $t_{\alpha\gamma}$, it does so much more drastically. A likely reason for the more drastic effect of vicinity total is that it gives a larger bonus to documents and therefore dominates small values of α.

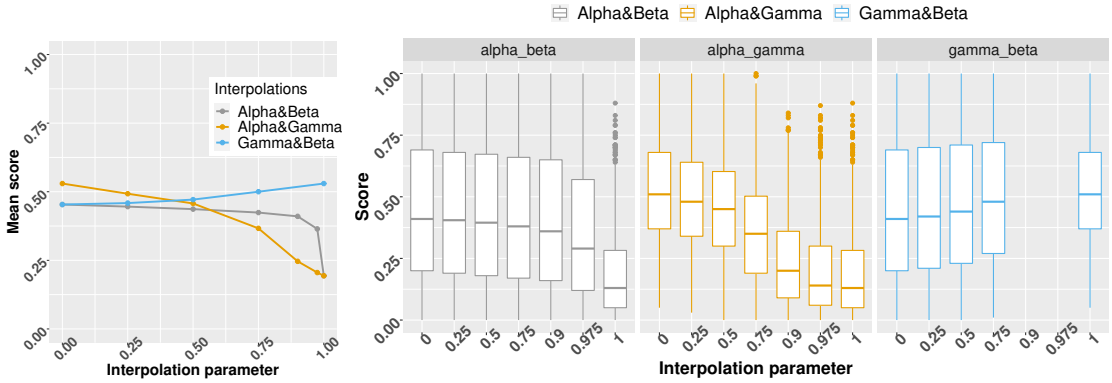

Figure 5.3 – *The scores over of interpolations between main variables. Each interpolation is a pairwise combination of the three main variables, while keeping the respective third variable at zero. It takes values from the variable written first in the label. Left: the mean scores for each interpolation. Right: boxplots for each interpolation. The two vacancies in the right panel are due to that extra simulations was done only to test combinations with high values of α and low values of β and γ. There are therefore no corresponding combination in the data set between only β and γ.*

The right part of Figure 5.3 contains the boxplots showing the spread of values around the mean scores. The most important observation here is including the vicinity total leads to large variance. This can be seen both in that the spread for $t_{\alpha\beta}$ is larger than for $t_{\alpha\gamma}$ and that the spread drops when $t_{\alpha\beta} = 1$ or $t_{\gamma\beta} = 1$ since $\beta = 0$ in both cases.

5.1.4 New single main variable

For the rest of the analysis, the discussion will focus on the interpolations $t_{\alpha\gamma}$. Shifting the focus to the trade-off between the different types of scoring is more interesting since it better illustrates the effect of the modification. While it is possible to create interpolations between all three of α, β, and γ, limiting the trade-offs to two of the variables makes the result easier to intuitively interpret.

One reason for only selecting one trade-off to focus on is that its influence on the score will be clearest if it is included alone in the model. Another reason is that the three different trade-offs are somewhat collinear. This means that the interpolations have overlapping effects on the score, with the influence of $t_{\alpha\beta}$ and $t_{\alpha\gamma}$ as an example. The difference between collinearity and interaction effects is that in interaction the factors affect each other's influence while in collinearity they have the same effect. This overlap makes it important to remove collinearities from the model.

Finally, selecting either $t_{\alpha\beta}$ or $t_{\alpha\gamma}$ is more suitable, since they represent the trade-off between the original score and a type of modification. Between these two, $t_{\alpha\gamma}$ has the largest span in mean scores, the more gradual change in mean score, and the lesser variance. It is, therefore, better from an applied perspective. Importantly, $t_{\alpha\gamma}$ also fit the assumption behind the Linear Mixed Model much better, even when the $t_{\alpha\beta}$ variable is transformed to level out the drastic change in scores for values of $t_{\alpha\beta}$ close to one.

As part of the investigation into $t_{\alpha\gamma}$, the statistical significance of including it as a fixed effect and as a random effect was tested using the likelihood ratio test. First the inclusion of t as only a fixed effect was tested. That is, a model with a fixed intecept, a fixed t, and a random intercept was tested against the nested model. The smaller model only included a fixed intercept and a random intercept. When comparing the two models, the likelihood ratio test accepts the inclusion of t ($\chi^2 = 846, p =< 1e-16$). The p-value given by the test says that it is very unlikely to get as an extreme difference between the two models if the extra variable included in the full model was not important.

When the likelihood ratio test is applied here and further down, the values of $t_{\alpha\gamma}$ are limited to $\{0, 0.25, 0.5, 0.75\}$ unless otherwise specified. This is done since it these values that are used when fitting the model.

Next, after establishing that the fixed influence of t is significant, the inclusion of t as a random variable is tested. When comparing the full model, where t is included both as fixed and random, with the nested model where t is only fixed, the likelihood ratio test accepts the larger model ($\chi^2 = 549, p =< 1e-16$).

5.1.5 The variable d

The variable d represents the parameter used in the Rank Biased Measure to control the bias towards high ranked entries in the lists. A larger value of d means that a larger number of entries of the ranked lists are relevant. Although the variable d can take continuous values in [0,1), it is best thought of as having discrete levels. The corresponding parameter does not have a linear function in the measure and it is, therefore, confusing to treat the variable as linear in the Linear Mixed Model. Furthermore, while there might be a linear relationship between the values of d and the score in the data set used here, this is unlikely to be true for more extreme values of d.

The left panel of Figure 5.4 shows the mean score over $t_{\alpha\gamma}$ for each value of d. The different levels of d have a clear impact on the mean score. The upper left panel also indicates that there is a slight interaction effect between d and the interpolation, which shows in that the span between the different levels of d decreases with larger values of $t_{\alpha\gamma}$.

The right panel of Figure 5.4 shows separate boxplots of scores for each $t_{\alpha\gamma}$ and d. Apart from the already mentioned difference in means for different values of d, the spread decreases with d. An exception is for high values of $t_{\alpha\gamma}$, where the trend is re-

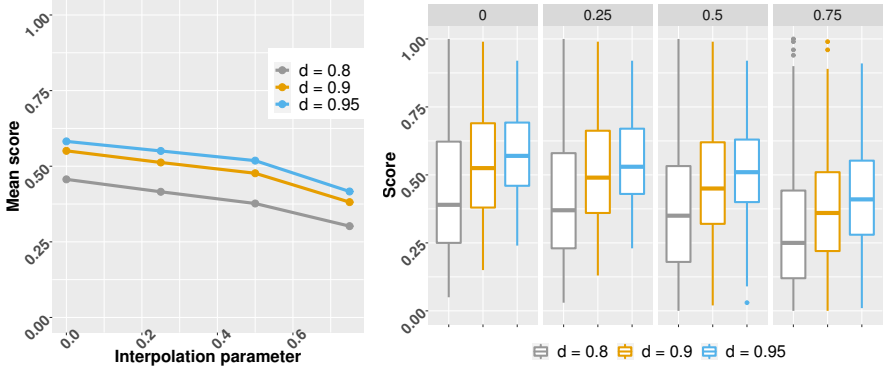

Figure 5.4 – *The influence of the depth variable d on the score for each value of $t_{\alpha\gamma}$. The different values of d are separated by different colors. Left: mean score over $t_{\alpha\gamma}$. Right: boxplots of scores for each $t_{\alpha\gamma}$.*

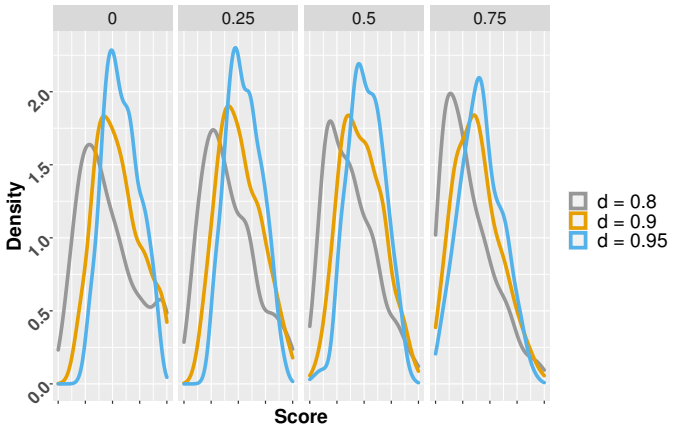

Figure 5.5 – *Distribution of scores for values of $t_{\alpha\gamma}$ and of d. The different values of d are separated by different colors.*

versed. A possible reason for this is that when the ranking is dominated by the original scoring the emphasis is on a smaller part of the ranked lists lessens the chance for spurious connections between documents.

In Figure 5.5, the density of the scores are displayed for each $t_{\alpha\gamma}$ and each d. As can be seen, there is a large influence of $t_{\alpha\gamma}$, with high values leading to density peaks close to scores at zero and lower values of $t_{\alpha\gamma}$ being more normally distributed around the scores at 0.5. The different values of d do however not have any obvious impact aside from the already mentioned changes in spread and mean.

From the above results, the variable d appears to be important for predicting the score. That the inclusion of d is significant is supported by the likelihood ratio test ($\chi^2 = 1417, p =< 1e-16$), when comparing two applications of the relevant Linear Mixed Models with and without d.

The importance of the interaction between d and $t_{\alpha\gamma}$ is not as clear from the figures. The likelihood ratio can again be applied by comparing a full model which adds both d and its interaction with $t_{\alpha\gamma}$ against a nested model that only adds d. The test either accepts or rejects the inclusion of the interaction depending on which data set that is used. The test accepts the full model for the data set used in this first part of the analysis (more correctly, the test rejects that the interaction is zero, which implies that the full model

should be used). However, when the test is applied to the larger data set used to fit the final model, it only accepts the inclusion of the interaction if $t_{\alpha\gamma}$ includes high values such as $\{0.9, 0.95, 1\}$ and otherwise rejects it ($\chi^2 = 2.3, p = 0.3$). Since those high values are not included when fitting the final model, neither is the interaction between d and $t_{\alpha\gamma}$. Dropping the weak interaction between $t_{\alpha\gamma}$ and d makes the final model easier to interpret and less likely to overfit.

5.1.6 The variable l

The variable l represents the number of words in the queries used in the simulations. The upper left panel of Figure 5.6 shows the mean score over $t_{\alpha\gamma}$ for each value of l, while the upper right panel shows the corresponding boxplots. As can be seen, neither the means nor the spread of the score is influenced much by the query length.

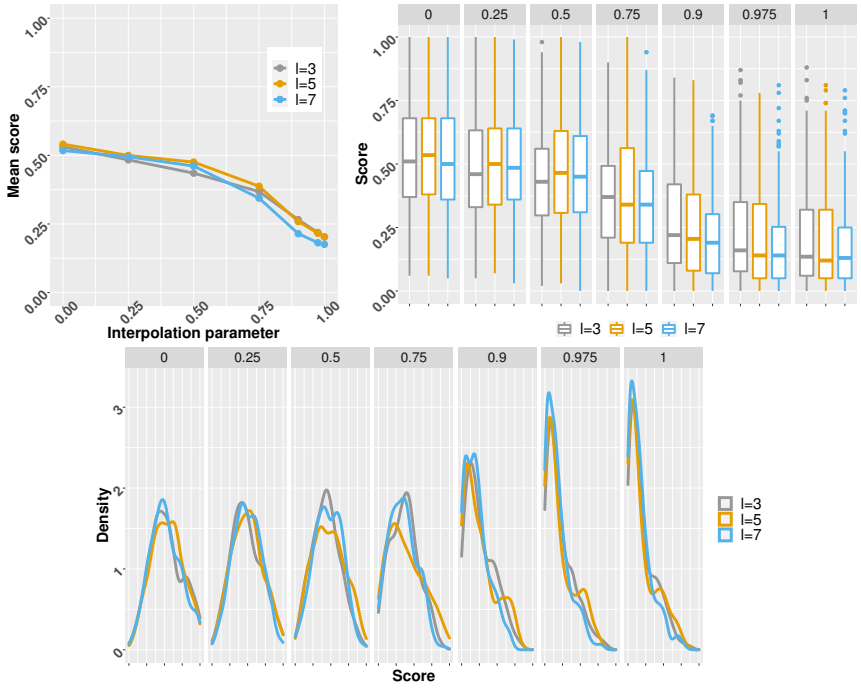

Figure 5.6 – *The influence of l on the score for each value of $t_{\alpha\gamma}$. The different values of l are separated by different colors. Top left: mean score over $t_{\alpha\gamma}$. Top right: boxplots of scores for each $t_{\alpha\gamma}$. Bottom: the distribution of scores for each $t_{\alpha\gamma}$.*

The lower panel of Figure 5.6, shows the distribution of the scores for $t_{\alpha\gamma}$ and for l. As with the mean and spread, l do not have any obvious influence.

Applying the likelihood ratio test leads to rejecting the inclusion of l ($\chi^2 = 1e-04, p > 0.9$) when comparing a relevant Linear Mixed Model with l versus a model without l. The test also rejects adding only an interaction between $t_{\alpha\gamma}$ and l ($\chi^2 = 0.03, p = 0.9$) . Unlike the rejection of the interaction between d and $t_{\alpha\gamma}$, the rejection of any inclusion of l do not depend on the values of $t_{\alpha\gamma}$ nor on which of the two data sets that are used.

5.2 Linear Mixed Model analysis

This section formulates the Linear Mixed Model, tests its assumptions, and applies it to data generated by the simulations.

5.2.1 Model data set

For the application of the Linear Mixed Model, a new data set was generated with more observations corresponding to the interpolation $t_{\alpha,\gamma}$. The relevant variables for the model data set are displayed in Table 5.3. Although the variable l is not included in the description or the model, the length of the queries still varies between three, five, and seven words, since there was no reason to choose a single length.

Another difference between the exploratory data set used above and the model data set is that the value of $t_{\alpha\gamma}$ is there restricted to $\{0, 0.25, 0.5, 0.75\}$. The observations corresponding to high values of $t_{\alpha\gamma}$ do not fit well with the assumptions behind the Linear Mixed Model and removing them, therefore, improves the model fit.

Table 5.3 – *Variables in the model data set. The variable s represents the scores, q represents the randomly generated queries, the variable d corresponds to the parameter with the same name used in the measure, and the main variable, $t_{\alpha\gamma}$, represent an interpolation between the original BM25 ranking score and each document's average vicinity score.*

Variables	Values	Total
s	$\in [0, 1]$	7152
q	$\in \mathbb{N}$	596
d	$\{0.8, 0.9, 0.95\}$	
$t_{\alpha,\gamma}$	$\{0, 0.25, 0.5, 0.75\}$	

The intraclass correlation coefficient and the effective sample size for the model data set are shown in Table 5.4. Compared to the exploratory data set, the intraclass correlation coefficient drops with 0.15. This means that the variance within each query's observations accounts for 15 percentage points more in the model data set. This also shows in that the effective sample size is a much larger fraction of the total number of observations. The variance reduction comes from the β variable had a larger variance than both α and γ (see Figure 5.3).

Table 5.4 – *The intraclass correlation coefficient and effective sample size for the model data set.*

Type	Sigma	ICC	Total
Independent clusters (var. q)	0.03	0.61	596
Residual	0.02	0.38	
Sample Size			7152
Effective Sample Size			922

5.2.2 Model formulation

Here the Linear Mixed Model that is used to describe the relationship between the selected independent variables and the score is formulated.

The dependent variable predicted by the model is the Rank Biased Cluster score. In the model, this variable is denoted with y, and all the scores relating to the ith query are

denoted with the vector $\mathbf{y}_i$. The independent variables are divided into fixed and random factors, where fixed consists of $t_{\alpha\gamma}$ and the three different levels of d. Their values for the ith query are written as the vectors $\mathbf{t}_{\alpha\gamma_i}$, $\mathbf{1}_i$, $\mathbf{d_{0.9}}_i$, and $\mathbf{d_{0.95}}_i$, with $\mathbf{1}_i$ being a vector of ones corresponding to the default level of d ($d = 0.8$). This means that the relationships between the dependent variable and $d_{0.9}$ or $d_{0.95}$ capture the influence of changing the value of d from the default $d_{0.8}$. An example is that if the default $d = 0.8$ gives a score at 0.5 and the model estimates the fixed effect of $d_{0.9}$ to 0.1, the actual estimated value for $d_{0.9}$ is $0.5 + 0.1 = 0.6$.

The main random factor in the model is the variable q, representing the randomly generated queries. There is an interaction between the random queries and $t_{\alpha\gamma}$, as shown in the first part of the analysis. This interaction means that the effect of $t_{\alpha\gamma}$ on the score varies around the value of its fixed part according to its random part.

With these considerations in mind, let q_i be the ith query, y_i be the observations from the ith query, and x_i and z_i be the corresponding values of the fixed and random independent variables. The value of the q and the default level of d is always at one and written with vectors of ones, i.e. with $\mathbf{1}_i$. For the vector $\mathbf{d_{0.9}}_i$, the jth element equals one if the jth d equals 0.9 and otherwise the element is zero. The values in the vector $\mathbf{d_{0.95}}_i$ are analogous. The model is formulated as

$$
\begin{aligned}
y_i &= x_i\lambda + z_i\eta + \varepsilon_i, \\
x_i &= \begin{bmatrix} \mathbf{1}_i & (d_{0.9})_i & (d_{0.95})_i & (t_{\alpha\gamma})_i \end{bmatrix}, \\
z_i &= \begin{bmatrix} \mathbf{1}_i & (t_{\alpha\gamma})_i \end{bmatrix}, \\
\eta &\sim \mathcal{N}(\mathbf{0}, \Sigma) \\
\varepsilon_i &\sim \mathcal{N}(\mathbf{0}_i, \sigma^2 I_i),
\end{aligned}
\tag{5.1}
$$

where λ, Σ, and σ^2 are estimated when fitting the model to the data.

5.2.3 Model assumptions

There are five assumptions behind the Linear Mixed Model, which all can be inferred from Equation (5.1). They are:

- the observations (scores) for each query are multivariate normally distributed,
- the observations for each query are independent of observations from other queries,
- the means of the multivariate normal distributions have a linear relationship with the fixed factors (d and $t_{\alpha\beta}$),
- there is no trend in the residuals (differences between actual observations and values predicted by the model),
- the error term (variance not explained by the multivariate normal distribution) is independent of other observations and drawn from a normal distribution with zero mean and a fixed variance.

According to Wiley and Wiley, there is no good test for the independence between scores in different queries but the four others can be assessed visually using diagnostic plots [43]. The relevant diagnostic plots are shown in Figure 5.7.

The assumption that the scores come from the expected multivariate normal distribution and the individual error term is independently and normally distributed can be assessed together by examining the distribution of the residuals. The residuals represent the deviation from the predicted mean and should, if the scores are normally distributed

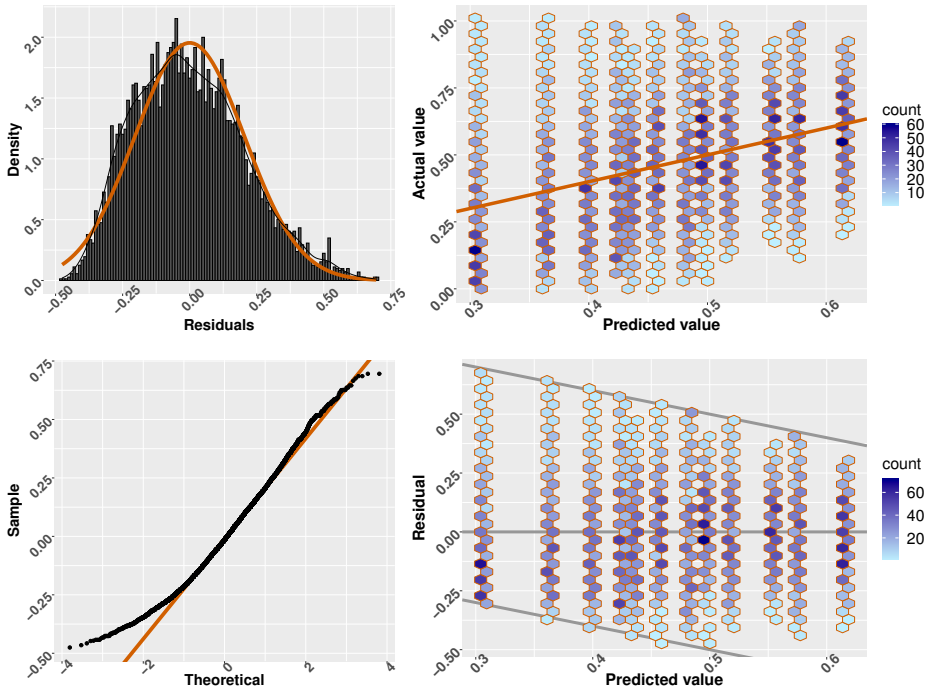

Figure 5.7 – *Diagnostic plots for the Linear Mixed Model. Top left: the density of the residuals, with the orange curve representing a corresponding normal distribution. Top right: the actual values over the predicted values, with the straight line representing a perfect match. Lower left: a QQ-plot, contrasting the residual scores against the normal distribution. Lower right: the residuals over the predicted values, with the diagonal lines representing the maximal residual for that predicted value.*

around its random mean, be normally distributed around zero. The upper and lower left panels of Figure 5.7 show the distribution of the residuals and a corresponding QQ-plot. As can be seen, the normal distribution is not a perfect fit. However, as the QQ-plot indicates, this is mainly due to there being too few large negative residuals. This might be an effect of that the density of scores does not approach zero for low scores. The model assumes that the scores take values from the real line and therefore fails to deal properly with the lack of negative scores.

The homogeneity of the residuals, i.e. that there is no trend in the residuals, can be assessed visually in the lower right panel of Figure 5.7, where the residuals are plotted for the predicted values. The grey diagonal lines in the figure represent the maximum and minimum possible residuals, given that the scores are restricted to between zero and one. On average, there is no trend between the predicted values and the residuals. An exception is a somewhat larger density in the lower-left corner, which likely is due to the already discussed discrepancy between the normality assumption and that restriction of the scores within zero and one.

The linear relation between the fixed factors and the mean scores can be assessed visually in the upper right panel of Figure 5.7, which shows the actual scores over the corresponding prediction by the model described in Equation (5.2). Although the figure shows that the scores vary a lot around the linear prediction, most of the data spread out symmetrically around the linearly increasing mean, represented in the figure by the solid line. A discrepancy to notice is the density of scores in the lower-left corner. A possible

reason for this might again be that the scores are cut off at zero, limiting the room for negative variation.

In summary, the assumptions fit the data in general, with the exception probably caused by the limiting of the score to between zero and one. This problem is discussed with the reasons for choosing the Linear Mixed Model in Section 3.4 and deemed acceptable given the aim and constraints of the thesis.

5.2.4 Final Model

The final model is reached by fitting the model specification to the model data using the R lme4 function [3]. The resulting model is described by

$$\begin{aligned} y_i &= \Big(0.48\mathbf{1}_i + 0.09(d_{0.9})_i + 0.13(d_{0.95})_i - 0.24(t_{\alpha\gamma})_i\Big) + \Big(\eta_1\mathbf{1}_i + \eta_2(t_{\alpha\gamma})_i\Big) + \varepsilon_i, \\ \eta &\sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0.04 & -0.02\\ -0.02 & 0.06\end{bmatrix}\right), \\ \varepsilon_i &\sim \mathcal{N}(\mathbf{0}_i, 0.006I_i), \end{aligned} \tag{5.2}$$

where i represents the ith query, the first set of large parenthesis denotes the fixed effects and the second denotes the random, and ε_i is the randomness not explained by the model.

Starting with the variable d, $0.48\mathbf{1}_i$ means that the default level of d at 0.8 sets the baseline score at 0.48. If d instead equals 0.9 for the jth observation within the ith query, the corresponding element $(d_{0.9})_{ij}$ then equals 1 adding 0.09 to the baseline score. Similarly, if d equals 0.95, 0.13 is added.

The fixed effect of the variable $t_{\alpha\gamma}$ is estimated to decrease the score with -0.24 when $t_{\alpha\gamma}$ is increased with one. This means that the model predicts that if $t_{\alpha\gamma}$ is changed from 0.5 to 0.75 the score will on average decrease with $-0.24 \times (0.75 - 0.5) = 0.06$.

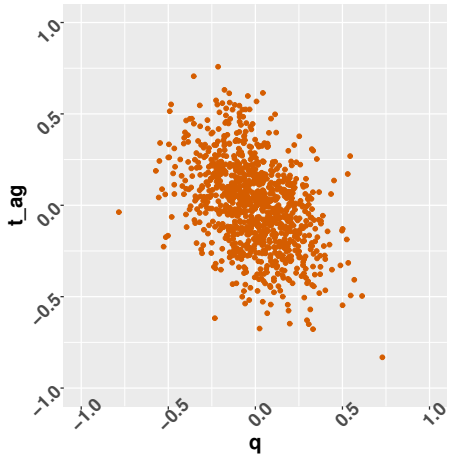

Figure 5.8 – *Scatterplot of 1000 samples from the distribution of the random coefficient vector η.*

The random effects consists of random changes to the mean, captured by η_1, and to the influence of $t_{\alpha\gamma}$, captured by η_2, for each query. The two random changes are described by the distribution of η, detailed in the Equation (5.2), and illustrated in Figure 5.8 where 1000 samples from the distribution are shown. This means that on top of the fixed influence common to all queries, the scores in each query has an individual addition of $\eta_1\mathbf{1}_i + \eta_2(t_{\alpha\gamma})_i$. As an example, η might for some query take the value $[0.15, -0.15]$.

Then $0.15 \times \mathbf{1}_i$ adds 0.15 to the mean for all scores from that query. When it comes to $t_{\alpha\gamma}$, its impact will be a combination of the fixed and the random part, so a unit change in $t_{\alpha\gamma}$ impacts the score with $(-0.24) + (-0.15) = -0.39$.

Finally, the error term ε_i adds a random error independently and identically generated for each score with the variance 0.006.

A limitation with the model is that if the random terms and the individual random error take somewhat extreme values, the model can predict scores outside the interval [0,1]. This is due to the previously noted discrepancy between the model assumption that the scores can take any real value and the scores actual interval [0,1].

6

Discussion

The main result is that varying the variables in the modified ranking algorithm tangibly and statistically significantly influences the interconnectedness in the ranked lists, although there is a large variance between different queries.

A related result is that using the vicinity average score, represented by the variable γ, lead to higher connectedness than the vicinity total, represented by the variable β. Furthermore, the average also leads to lower variance compared to expansion without normalisation. A possible explanation of these differences is that articles with a mix of relevant and irrelevant citation links are penalized when using the average. For queries where the top documents have similar scores and where there are many connections in the relevant part of the list, using the vicinity average may lead to that documents only rises high up in the ranking if they are connected to many top hits. This is likely to happen for a few documents, but these will also be more connected. On a similar track, for queries where there are large differences in the score between the top-ranking documents or where there are few connections in the relevant part of the list, using a total vicinity score might lead to that all documents related to the top document(s) dominate the reranking. That being said, it is not certain why this is the case and it would have been interesting to look further into the differences between the total and average vicinity scores.

The results also show that the depth variable, d, significantly determine how connected the lists are judged to be, with a larger d leading to a higher degree of interconnectedness. That there is a significant difference relating to different values of d is expected, since changing between the three different values of d leads to a large difference in the size of the influential part of the lists. It is, however, surprising that increasing d correlated with more connections in the lists. A possible reason for this is that the articles in the top of the ranked lists often belonged to clusters but that the connections were first revealed a bit further down.

Another also somewhat surprising result was the insignificance of the query specificity, especially given the significance of the topic. The expectation was that specific queries would return articles from more specific topics, which would then increase the connections between high ranking articles, while vague queries would retrieve articles from a broad range of topics, leading to few connections. The different amount of connections would then, in turn, lead to different impact of the ranking algorithm modification. A suspected reason for why the query specificity was not important is a deficiency in the implementation. The randomly generated query terms are processed with stop word lists, removing common words like 'and', 'or', 'a' etc., but they do not limit the query to a single language. While English is the dominating language for the sampled query terms, with about three out of four articles being in English, other languages such as Chinese, French, German etc., are relatively common as well. It is therefore likely that

longer queries will include terms from different languages and since the titles of articles are usually only written in one language, the query simply becomes an integration of two or more disjoint queries. The result is therefore that a longer query, in the cases where terms from different languages are included, creates a broader query instead of a more specific. A possible partial solution would have been to limit the query terms to letters in the Latin alphabet, which was not implemented due to the time constraints. The problem would, however, have remained for the different European languages, with no obvious solution. One attempt would have been to match titles against a list of English stop words (e.g. 'and') and only sample from the titles that match.

7
Conclusion

This thesis aims to determine if the ranked articles' interconnectedness can be controlled by three variables weighting together the default ranking with the ranking according to the vicinity's total and average scores. The analysis shows that the variables can influence the interconnectedness. Furthermore, it shows that it is enough to replace the three variables with an interpolation between two of the three.

The point with this aim was to showcase an indirect approach to studying exploratory searching. In this thesis, the focus on *how* the search engine is affected enabled developing the Rank Biased Cluster-measure. This, in turn, enabled using simulations. Being able to use simulations is helpful for a small field since they are convenient and free. In this thesis, simulations also enabled generating enough data to handle the large variance that the random queries introduce and to account for a few interactions in the analysis. Preferably, many more interactions would have been included and in a larger study, that can be straightforwardly incorporated by simply extending the simulations. This shows that the component-wise analysis argued for by Ferro and Silvello can be applied in the field of Exploratory Search.

If the indirect approach is interesting depends on the context. It is probably hard to get the benefits of the approach when studying graphical user interfaces, which is a common focus in the field Exploratory Search. Furthermore, the results do not show if one type of ranking is better than another. That being said, the results can still be useful when designing search engines for exploratory search. In the case of the modifications examined in the thesis, it might support exploratory searching by helping well-connected documents rise in the ranking. This can be seen as a first compilation of the top results and their most relevant references, and automates a typical step in exploring academic topics, where the user manually goes through the references and citations of the top results. Although more research is required for reaching a good understanding of the modification, this study indicates that the indirect approach can lead to results useful for exploratory searching.

7.1 Suggestions for future work

This study leaves many questions open for further pursuit. A few suggestions are given here.

The implementation used in this study is limited in many ways, due to the constraints in the thesis. There are many settings and features that could be set and potentially influence the results. It would, for example, be interesting to examine many more interactions between the ranking algorithm modification and aspects of the search engine set-up to better support the conclusions.

When it comes to the analysis of the ranking algorithm modification, a more thorough investigation is needed for a clear picture. The Rank Bias Cluster-measure only captures part of the picture of the effect of the modifications on the connectedness of the lists. It would, for example, be valuable to look into the characteristics of the articles that climbs or descends. It would also be interesting to compare high scores given by the Rank Biased Cluster measure with a similarity score of the highly ranked documents. Preferably, the latter could be calculated from the abstracts that were not considered by the BM25 algorithm in the simulations.

The ranking algorithm modification itself could be changed in many ways. One interesting option is to investigate how differentiating between citations and references effect the connectedness. Another is to try more advanced forms of citation expansion, such as the popular inclusion of citation context.

The application of the Linear Mixed Model had, as expected, difficulties with the limited interval for the scores from the measure. It would be interesting to compare how robust the Linear Mixed Model is for this problem by comparing the model with for example Generalized Linear Mixed Models with either a Poisson distribution or a Binomial distribution, as well as with the Beta regression.

Finally, the perhaps most interesting suggestion is to perform similar studies using knowledge graphs instead of citation links. Knowledge graphs aim to support searchers by linking relevant information not directly captured by the query. An example of its use is complementing a book search with a link to the authors Wikipedia site. It is currently a hot field of research and is paid attention to from companies such as Google and Microsoft.

Bibliography

Book Sources

[9] William Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines - Information Retrieval in Practice*. Pearson Education, Inc, 2015.

[11] E. Demidenko. *Mixed models : Theory and applications with r*. Somerset: John Wiley & Sons, Incorporated, 2013. URL: https://ebookcentral.proquest.com.

[28] Peter Olofsson and Mikael. Andersson. *Probability, statistics, and stochastic processes*. 2nd ed. Wiley, 2012. ISBN: 9780470889749.

[33] Andrew Rutherford. *ANOVA and ANCOVA : A GLM Approach*. second. John Wiley & Sons, Incorporated, 2011.

[43] M. Wiley and J. Wiley. *Advanced R Statistical Programming and Data Models*. Apress, Berkeley, CA, 2019.

Other Sources

[1] Waleed Ammar et al. "Construction of the Literature Graph in Semantic Scholar". In: (May 2018). URL: http://arxiv.org/abs/1805.02262.

[2] Kumaripaba Athukorala et al. "Is exploratory search different? A comparison of information search behavior for exploratory and lookup tasks". In: *Journal of the Association for Information Science and Technology* 67.11 (Nov. 2016), pp. 2635–2651. ISSN: 23301643. DOI: 10.1002/asi.23617.

[3] Douglas Bates et al. "Fitting linear mixed-effects models using lme4". In: *Journal of Statistical Software* 67 (2014), pp. 1–48.

[4] Joeran Beel and Bela Gipp. "Google Scholars Ranking Algorithm: An Introductory Overview". In: *Proceedings of the 12th International Conference on Scientometrics and Informetrics* 1 (2009), pp. 230–241. URL: www.docear.org.

[5] Joeran Beel et al. "Research-paper recommender systems: a literature survey". In: *International Journal on Digital Libraries* 17.4 (Nov. 2016), pp. 305–338. ISSN: 14321300. DOI: 10.1007/s00799-015-0156-0.

[6] Christopher W. Belter. "A relevance ranking method for citation-based search results". In: *Scientometrics* 112.2 (Aug. 2017), pp. 731–746. ISSN: 15882861. DOI: 10.1007/s11192-017-2406-y.

[7] Benjamin A. Carterette. "Multiple testing in statistical analysis of systems-based information retrieval experiments". In: *ACM Transactions on Information Systems* 30.1 (Feb. 2012). ISSN: 10468188. DOI: 10.1145/2094072.2094076.

[8] Chaomei Chen. *Cascading Citation Expansion*. Tech. rep. URL: https://i4oc.org/.

[10] Jonathan Demelo, Paul Parsons, and Kamran Sedig. “Ontology-Driven Search and Triage: Design of a Web-Based Visual Interface for MEDLINE”. In: *JMIR Medical Informatics* 5.1 (Feb. 2017), e4. DOI: 10.2196/medinform.6918.

[12] Jacob C. Douma and James T. Weedon. “Analysing continuous proportions in ecology and evolution: A practical introduction to beta and Dirichlet regression”. In: *Methods in Ecology and Evolution*. Vol. 10. 9. British Ecological Society, Sept. 2019, pp. 1412–1430. DOI: 10.1111/2041-210X.13234.

[13] David Dubin. *The Most Influential Paper Gerard Salton Never Wrote*. Tech. rep. 4. 2004, pp. 748–764.

[14] Elasticsearch. “About | Elastic”. In: (). URL: https://www.elastic.co/about/.

[15] Elasticsearch. *Information out: search and analyze | Elastic Reference [7.3] | Elastic*. URL: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-analyze.html.

[16] Elasticsearch. *Pluggable Similarity Algorithms | Elasticsearch: The Definite Guide [2.x] | Elastic*. URL: https://www.elastic.co/guide/en/elasticsearch/guide/2.x/pluggable-similarites.html.

[17] Elasticsearch. “Similarity | Elasticsearch Reference [7.3] | Elastic”. In: (). URL: https://www.elastic.co/guide/en/elasticsearch/reference/7.3/similarity.html.

[18] Elasticsearch. *Similarity module | Elasticsearch Reference [7.3] | Elastic*. URL: https://www.elastic.co/guide/en/elasticsearch/reference/7.3/index-modules-similarity.html.

[19] Masaki Eto. “Evaluations of context-based co-citation searching”. In: *Scientometrics* 94.2 (2013), pp. 651–673. ISSN: 01389130. DOI: 10.1007/s11192-012-0756-z.

[20] Masaki Eto. “Rough co-citation as a measure of relationship to expand co-citation networks for scientific paper searches”. In: *Proceedings of the Association for Information Science and Technology* 53.1 (2016), pp. 1–4. ISSN: 23739231. DOI: 10.1002/pra2.2016.14505301131.

[21] Nicola Ferro and Gianmaria Silvello. “Toward an anatomy of IR system component performances”. In: *Journal of the Association for Information Science and Technology* 69.2 (Feb. 2018), pp. 187–200.

[22] Jiafeng Guo et al. “A Deep Look into neural ranking models for information retrieval”. In: *Information Processing and Management* (2019). ISSN: 03064573. DOI: 10.1016/j.ipm.2019.102067.

[23] A. Cecile J.W. Janssens and M. Gwinn. “Novel citation-based search method for scientific literature: application to meta-analyses”. In: *BMC Medical Research Methodology* 15.1 (Oct. 2015). ISSN: 14712288. DOI: 10.1186/s12874-015-0077-z.

[24] Shah Khalid et al. *On The Current State of Scholarly Retrieval Systems*. Tech. rep. 1. 2019, pp. 3863–3870. URL: www.etasr.com.

[25] Shengbo Liu et al. “Literature retrieval based on citation context”. In: *Scientometrics* 101.2 (Nov. 2014), pp. 1293–1307. ISSN: 0138-9130. DOI: 10.1007/s11192-014-1233-7. URL: http://link.springer.com/10.1007/s11192-014-1233-7.

[26] Shixia Liu et al. “A survey on information visualization: recent advances and challenges”. In: *Visual Computer* 30.12 (2014). ISSN: 01782789. DOI: 10.1007/s00371-013-0892-3.

[27] Gary Marchionini. “Exploratory search: from finding to understanding”. In: *Communications of the ACM* 49.4 (2006), pp. 41–46.

[29] *Pluggable Similarity Algorithms | Elasticsearch: The definitive Guide [2x] | Elastic*. URL: https://www.elastic.co/guide/en/elasticsearch/guide/current/pluggable-similarites.html.

[30] *Practical BM25 - Part 2: The BM25 Algorithm and its Variables | Elastic Blog*. URL: https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables.

[31] Sabir Ribas et al. “Simplified Relative Citation Ratio for Static Paper Ranking: UFMG/LATIN at WSDM Cup 2016”. In: (Mar. 2016). URL: http://arxiv.org/abs/1603.01336.

[32] Stephen Robertson. “A brief history of search results ranking”. In: *IEEE Annals of the History of Computing* 41.2 (Apr. 2019), pp. 22–28. ISSN: 19341547. DOI: 10.1109/MAHC.2019.2897559.

[34] Tetsuya Sakai. “Statistical significance, power, and sample sizes: A systematic review of SIGIR and TOIS, 2006-2015”. In: *SIGIR 2016 - Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Inc, July 2016, pp. 5–14. ISBN: 9781450342902. DOI: 10.1145/2911451.2911492.

[35] G. Salton, A. Wong, and C. S. Yang. “A Vector Space Model for Automatic Indexing”. In: *Communications of the ACM* 18.11 (Nov. 1975), pp. 613–620. ISSN: 15577317. DOI: 10.1145/361219.361220.

[36] Esben S{\o}rig et al. “Evaluation of Rich and Explicit Feedback for Exploratory Search”. In: 2019, pp. 309–315. ISBN: 9781450367110. DOI: 10.1145/1122445.1122456.

[37] K. Sparck Jones, S. Walker, and S. E. Robertson. “Probabilistic model of information retrieval: Development and comparative experiments. Part 2”. In: *Information Processing and Management* 36.6 (Nov. 2000), pp. 809–840. ISSN: 03064573. DOI: 10.1016/S0306-4573(00)00016-9.

[38] Peter D. Turney and Patrick Pantel. “From frequency to meaning: Vector space models of semantics”. In: *Journal of Artificial Intelligence Research* 37 (Jan. 2010), pp. 141–188. ISSN: 10769757. DOI: 10.1613/jair.2934.

[39] Nicolas Vibert et al. “Effects of domain knowledge on reference search with the PubMed database: An experimental study”. In: *Journal of the American Society for Information Science and Technology* 60.7 (July 2009), pp. 1423–1447. ISSN: 15322882. DOI: 10.1002/asi.21078.

[40] Alex D. Wade et al. “WSDM Cup 2016 - Entity ranking challenge”. In: *WSDM 2016 - Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, Inc, Feb. 2016, pp. 593–594. ISBN: 9781450337168. DOI: 10.1145/2835776.2855119.

[41] William Webber, Alistair Moffat, and Justin Zobel. “A similarity measure for indefinite rankings”. In: *ACM Transactions on Information Systems* 28.4 (Nov. 2010). DOI: 10.1145/1852102.1852106.

[42] Ryen W. White and Resa A. Roth. "Exploratory Search: Beyond the Query-Response Paradigm". In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1.1 (Jan. 2009), pp. 1–98. ISSN: 1947-945X. DOI: 10.2200/s00174ed1v01y200901icr003.

[44] Jian Wu et al. *CiteSeerX: AI in a Digital Library Search Engine*. Tech. rep. URL: www.aaai.org.

[45] Chenyan Xiong, Russell Power, and Jamie Callan. "Explicit semantic ranking for academic search via knowledge graph embedding". In: *26th International World Wide Web Conference, WWW 2017*. International World Wide Web Conferences Steering Committee, 2017, pp. 1271–1279. ISBN: 9781450349130. DOI: 10.1145/3038912.3052558.