Florian Ziessler, BSc

A Framework for E-Learning in the Field of Information Retrieval

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme:
Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, September 2018

Florian Ziessler, BSc

Ein Framework für E-Learning im Bereich von Information Retrieval

Masterarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium:
Information and Computer Engineering

eingereicht an der

Technischen Universität Graz

Betreuer

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Institute of Interactive Systems and Data Science
Vorstand: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Graz, September 2018

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Graz, _____ _____

Date Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Dissertation identisch.

Graz, am _____ _____

Datum Unterschrift

Abstract

Nowadays e-learning is on everyone's lips. This is due to its versatility and flexibility as well as the technological progress. The possibility to learn independent of time and location is becoming more important. Furthermore, e-learning allows to adapt the subject matter to the individual. Companies, schools and universities start to use this kind of learning more and more either exclusively or in combination with traditional teaching forms. Especially in the STEM-domain (Science, Technology, Engineering, and Mathematics) it is necessary to provide support for students through learning platforms and additional tools.
This thesis presents ISRapp, a prototype for e-learning in the field of information retrieval. It aims to implement the concept of Motivational Active Learning (MAL). MAL was first applied in the year 2013 and is based on the idea to increase the participation and attention of students by means of motivating strategies in connection with game design. For this learning format the Moodle learning platform was used. It allows to track the progress of a student and responses with instant feedback, but it does not provide the possibility to experiment with the subject matter.
Findings based on literature research of related work and predefined requirements served as a foundation and guided the development. In the current state the developed tool consists of two parts. Part one allows the user to experiment with the essentials of a search engine, part two can be used to play with text-statistics. The ISRapp can be used with online and offline data sources. It is platform independent since it was written in Java. The evaluation of the prototype was divided into two stages related to the state of development of the two parts and led to a qualitative and a quantitative evaluation. However, the results of the evaluation are promising and positive. Students could handle both parts of the tool after a short introduction period without significant problems. Most of them felt motivated and could imagine to use such a tool also in other areas.

Nevertheless, the testing revealed weak spots and showed plenty options for improvement.

Kurzfassung

Heutzutage ist „e-learning“ in aller Munde. Das ist auf die Vielseitigkeit und Flexibilität dieser Lernform sowie auf den technologischen Fortschritt zurückzuführen. Die Möglichkeit ort- und zeitunabhängig zu lernen wird immer wichtiger. Weiters erlaubt es e-learning Lernmaterial an den einzelnen anzupassen. Firmen, Schulen und Universitäten fangen an diese Art von Lernen immer öfter in ihren Kurse zu integrieren. Kurse werden entweder ausschließlich mittels e-learning angeboten oder in Kombination mit konventionellen Lernmethoden. Im Speziellen in den MINT-Fächern (Mathematik, Informatik, Naturwissenschaft und Technik) ist es wichtig Studenten mit Lernplattformen und zusätzlichen Lernhilfen zu unterstützen.
In dieser Arbeit wird ISRapp vorgestellt. ISRapp ist ein Prototyp für e-learning im Bereich von Information Retrieval. Er zielt darauf ab das Konzept von „Motivational Active Learning“ (MAL) zu implementieren. MAL wurde im Jahr 2013 zum ersten Mal vorgestellt und basiert auf der Idee die Teilnahme und Aufmerksamkeit von Studenten mittels motivierenden Strategien in Verbindung mit einem Spiel-Design zu steigern. Für dieses Lernformat wurde Moodle als Lernplattform eingesetzt. Es ermöglicht den Fortschritt eines Studenten zu verfolgen und kann unmittelbar Rückmeldungen liefern jedoch fehlt die Funktion um mit dem gelernten zu experimentieren.
Resultate aus Literaturrecherche und vorgegebenen Anforderungen dienten als Grundlage für die Entwicklung der Applikation. Im aktuellen Zustand besteht der Prototyp aus zwei Teilen. Teil eins ermöglicht es dem Benutzer mit Grundlagen einer Suchmaschine zu experimentieren, Teil zwei kann verwendet werden um mit Text-Statistiken zu spielen. Die ISRapp kann mit lokalen Daten als auch mit Daten, die online verfügbar sind, verwendet werden. Sie ist plattformunabhängig da sie in Java geschrieben wurde. Die Evaluierung des Prototyps wurde, bezogen auf den Entwicklungsstand der einzelnen Teile, in zwei Abschnitte aufgeteilt. Dies führte

zu einer qualitativen und einer quantitativen Evaluierung. In beiden Fällen wiesen die Resultate auf ein vielversprechendes System hin. Von Teilnehmern der Evaluierung kamen zum überwiegenden Teil positive Rückmeldungen. Die Studenten konnten nach einer kurzen Phase der Eingewöhnung ohne große Probleme mit der Applikation umgehen. Die meisten Probanden hatten das Gefühl durch die ISRapp motivierter zu sein und konnten sich vorstellen eine Applikation dieser Art auch in anderen Bereichen zu verwenden. Dessen ungeachtet hat die Evaluierung aufgezeigt, dass der Prototyp noch einige Schwachstellen in sich birgt und ein enormes Potential an Verbesserungen vorhanden ist.

Contents

List of Figures

1 Introduction

1.1 Motivation

> *"Knowledge acquisition is no longer mainly restricted to classical institutions and formal learning (as in schools and universities) but is also connected to informal learning settings at home in leisure time or at the workplace."* (Hesse, 2009)

With the evolution of technology new possibilities arise. This includes many aspects in everyday life and also applies to the way we learn. Hesse (2009) pointed out that due to the development of Web 2.0 and its current state a significant change in learning takes place. Especially the emergence of social software and social networks create an alternative view on this topic. The result involves an interaction between formal and informal learning. Social software can be divided into three groups (Hesse, 2009):

- Social exchange between people (e.g. Facebook[1])
- Knowledge exchange (Bookmarking Systems like Delicious[2])
- Share knowledge (e.g. Wikipedia[3])

Even though each of the systems serve a special purpose, the lines may blur between each other. It includes the creation of knowledge as well as the use of it. Furthermore, using social software allows to share knowledge in a very easy manner. Besides tools from the area of social software a specialization on software especially for the purpose in the field of learning happened. Already Eklund, Kay, and Lynch (2003) saw a big potential and promised a tremendous rise of e-learning software in the following years. In the 1960s education started in a new era with the first system

[1] Facebook https://www.facebook.com/, accessed 2017-9-1
[2] Delicious https://del.icio.us/, accessed 2017-9-1
[3] Wikipedia https://www.wikipedia.org/, accessed 2017-9-1

for computer based training by the use of system called PLATO (Programmed Logic for Automated Testing) (Murphy & Appel, 1977; Bezovski & Poorani, 2016; Jones, 2015). This approach was a step in a new direction and triggered a slow but steady increase of use in other applications. In the 1980s the idea got a boost when the principle of e-learning started as an alternative to standard face-to-face teaching in the field of aviation (Brown & Fallon, 2002). However, it still took some time to increase its popularity. This particularly applies to the area of STEM-education (Science, Technology, Engineering, and Mathematics (Gonzalez & Kuenzi, 2012)). E-learning is suited for people that are not able to attend educational institutions either for personal reasons or locational reasons. This applies notably for developing countries that lack in teaching staff, especially for STEM-education. Pillay and Kappus (2015) discuss this problem by pointing out the importance of e-learning by conducting a case study for developing countries. The selected country for the study was an island in the pacific with a population of about 19000 and had 40 secondary schools. By the use of ICT (information computer technology) intervention (Pillay & Kappus, 2015) teaching and learning of STEM subjects should be improved. The application of technology in that area revealed two issues, namely on one side the lack of qualified STEM-teachers and on the other side the lack of quality teaching resources. It turned out that both problems could be tackled even if it initially took some time to embolden teachers as well as students to participate.
Pirker, Riffnaller-Schiefer, and Gütl (2014) show a study with university students in STEM-education. Their approach - Motivational Active Learning (MAL) - is based on Technology-Enabled Active Learning (TEAL) (Pirker, Gütl, Belcher, & Bailey, 2013) which is a learning format for teaching physics at MIT. This kind of e-learning methodology combines TEAL with engaging gamification techniques. It allows to keep students concentrated on a theoretical topic for hours due to its interactive manner that contains different activities such as research and discussion questions, concept questions, calculation questions or even short programming questions. The study was conducted in the the course “Information Search and Retrieval (ISR)” at the University of Technology in Graz, Austria in the winter semester 2013 in which 28 students participated. The course aims to impart basics in information searching and retrieval. That includes topics in theory as well as exercises to deepen mathematical and algorithmic ideas. Additionally the course relies on a e-leaning platform as assistance

system, which is used to support students with content and also to track the progress of students.
The goal of this thesis is to develop a prototype that extends and provides functionality which is not yet implemented, but is the idea of MAL. MAL's concept relies on the interactivity in order to increase the students attendance through motivational strategies with game-design connections (Pirker et al., 2014). Therefore, the intention of the prototype is to realize interactive experimentation with concepts in the field of information retrieval. In the following section the exact problem is specified.

1.2 Problem Definition

The study of Pirker et al. (2014) reveals that involving students in the teaching process keeps them a lot more motivated compared to simple face-to-face teaching. Even though the evaluation of the MAL-study shows the success of the model, this learning scheme still lacks in some parts as explained in the following.
Based on a university course related to information retrieval, a study regarding e-learning was conducted. In that course students were taught not just in a face-to-face manner but also motivated to participate in discussions. Furthermore practical parts and little research were included to maintain interactive behavior. To support the learning process, students were encouraged to work with the Moodle[4] learning management system in order to do their exercises. Moodle also provided content and additional material that were used during the classes.
The basic procedure when working with exercises contained two steps: Completing the exercise and getting a result. Depending on the type of the exercise, a response happened immediately after the completion or with some delay, if manual evaluation was needed. Due to the fact that students could check their results subsequently a high motivation was achieved. Also, a detailed feedback and the possibility to correct an exercise contributed to an increased participation. It turned out that this kind of computer aided learning combined with discussions was a great incentive in such a way that students did not just improve but also had fun

[4]Moodle https://moodle.org/, accessed 2017-5-10

trying out learned techniques. In the first attempt, MAL did not provide this functionality and therefore it had to be done by the use of external applications. This raised the need of a new system that implements the functionality of testing and experimenting with new topics interactively. Hence, the idea of an extension to the already existing system came up. It should allow to deepen learned concepts from the course "Information Search and Retrieval". The resulting application should allow to load data from external sources as well as being used with predefined data. Based on these prerequisites, this thesis focuses on a suitable solution to fulfill the requirements.

1.3 Structure of the Work

The following chapter 2 discusses the background knowledge that is used and implemented in the prototype of this thesis. This starts with some important definitions in the considered field and includes background in e-learning as well as information retrieval, since this is the topic that should be transferred by e-learning. The last part in this chapter concerns text statistics.

In chapter 3 related work in terms of e-learning is taken into account. This includes applications that are used in the area of information retrieval and also applications in different fields with similar usage. The gained insights complement the requirements for the implementation of the prototype.

Chapter 4 presents the practical implementation of the prototype. Based on a listing of the requirements, the concept and the architecture will be discussed. Afterwards the used technologies are listed. Subsequently, the concrete implementation will be explained in detail.

Chapter 5 describes the methodology of the conducted evaluation. In the first section an overview of the participants who took part in the evaluation is presented. The evaluation consisted of two parts, a quantitative and a qualitative evaluation. Both are structured the same way. The structure contains an explanation of the experimental setup and the procedure step and is concluded with a summary of the results.

The concluding chapter 6 summarizes the main findings. It discusses the problems of the prototype that were revealed in the course of the evaluation. Finally, the last section provides a future outlook.

2 Background

This chapter gives an overview of the main concepts that are used in the information retrieval course and are covered in the interactive learning prototype that was developed within this thesis. Those that are not, generally extensions of the basics, will be added in future versions or just serve as additional information. However, the intention of this thesis is to provide a tool that helps to understand and experiment with concepts of information retrieval. It should act as a playground and environment to experiment and deepen knowledge. Therefore also e-learning is a topic whose background needs to be taken into account. The methods presented by the tool can be divided into two parts. A practical part, that demonstrates the basics of a search engine and an analytical part that can be used to create statistics of text.
The following section will cover basics regarding e-learning starting with an introduction, followed by some historical background, continuing with an example learning management systems and ends with e-learning models. In the subsequent section, fundamentals of the information retrieval process will be explained. The final section in this chapter is about text statistics.

2.1 E-learning

The term "e-learning" finds many definitions in the literature. Taking the following two as an example, one can already get an impression of its meaning.

> *"E-learning is defined as the whole activities of teaching and learning based on computer management environments constructed from net-*

> *work information techniques with interactive communications."* (Yu & Fan, 2009)

> *"E-learning: Any learning, training or education that is facilitated by the use of well-known and proven computer technologies, specifically networks based on Internet technology."* (Brown & Fallon, 2002)

With e-learning an alternative way of learning comes into our life. Yu and Fan (2009) summarize this as a result of the progress and development of technology in the past few years that led to new possibilities in education. This especially affects the process of conventional teaching methods which can be extended or even replaced by e-learning models. Zhang, Zhao, Zhou, and Nunamaker (2004) discuss the question "Can e-learning replace classroom learning?" The article shows a study in which classroom teaching and e-learning were directly compared. Undergraduate students from ten different majors were randomly assigned to the experimental group and the default group. It turned out that the test scores of the e-learning group were considerable higher than in the group with ordinary classroom teaching methods. Based on a post questionnaire this was mostly due to the possibility of a self-controlled learning process and the interactive multimedia content. These results from 2004 already showed that e-learning can be a promising alternative to the classical learning approach. In another study Owens and Price (2010) try to answer the very similar question: "Is e-learning replacing the traditional lecture?" Therefore e-learning and IT as learning technologies are inspected on how to be used as an alternative to the classical way of teaching. The conclusion is based on experiments with students in higher education and suggests a combination of both, traditional learning methods and e-learning.
However, by the use of e-learning models the learning process can be personalized, learning is more flexible and portable so it can be exerted almost everywhere and is not anymore restricted to a location (Zhang et al., 2004). Compared to traditional classroom learning e-learning brings along both, advantages as well as disadvantages. A brief comparison is shown in table 2.1.

	Traditional Classroom Learning	E-Learning
Advantages	• Immediate feedback • Familiar to instructor and students • Motivating students • Development of social interaction	• Learner is in focus and autonomous • Independent of time and location - world-wide audience possible • Cost-effective for learners • Unlimited access to content, knowledge reuse and sharing
Disadvantages	• Instructor centered • Not independent of time and location • Higher effort for knowledge transfer	• No immediate feedback • More preparation effort for instructor • Not suitable for every student • May involve frustration, confusion or anxiety

Table 2.1: Comparison of traditional classroom learning and e-learning. Taken and adapted from Zhang, Zhao, Zhou, and Nunamaker (2004).

2.1.1 Historical Background

The first approach to e-learning already started in the 1960s with a system for computer based training system called PLATO (Programmed Logic for Automated Testing) (Murphy & Appel, 1977; Bezovski & Poorani, 2016; Jones, 2015). However, using technology as an aid in education has its

origin much earlier in time. The idea of e-learning goes back to the 1980s, where it began to take shape. This will be elucidated in the following by the use of the book of Brown and Fallon (2002) on which this section is based on. In the period of the 1980s the aviation industry became aware of the fact that it is essential to train the stuff with the most up-to-date information available in order to sustain safety (Brown & Fallon, 2002). Based on the needs of a more efficient way for information transfer aircraft manufactures saw in computer based training (CBT) a new alternative to deliver knowledge. This could be put down to four reasons presented in the following list taken and adapted from Brown and Fallon (2002).

- In comparison to printed learning material, CBT provides a significantly better training experience due to its media-rich interactive content.
- Through tracking and evaluation it can be assured that personnel meets the necessary standards.
- This kind of training is available 24 hours a day, 7 days a week.
- A particular training task can be accessed on demand.

Since this way of learning turned out to be very successful a few millions of dollars were invested on computer based teaching material in the following years. After the spreading of CBT in the aviation industry one problem came up. This was the result of different vendors - McDonnell-Douglas, Airbus, Boeing and others - that did not only create their own proprietary software but also made it only usable with their own proprietary hardware. In consequence of vendors going one's own way airlines had to use another setup for each type of airplane which of course caused high costs. To counteract this trend, the Aviation Industry CBT Committee (AICC) was founded and developed in 1993 the AICC CMI (computer-managed instruction) specification[1]. This is a guideline for interoperability between CMI systems with the focus on how to share data in local area networks using platforms from different vendors (Brown & Fallon, 2002). CMI systems form the basis of todays learning management systems (Brown & Fallon, 2002).

Through the introduction of standards for HTML by the World Wide

[1] AICC CMI https://www.immagic.com/eLibrary/ARCHIVES/TECH/AICC_US/A040816G.pdf, accessed 2017-11-14

Web Consortium[2] (W3C) and, of course the technological progress, a great impact on the development of e-learning was made (R. Jones, 2002). Hence, also the AICC CMI specification was adapted in order to supply guidance for web-based CMI (specification AGR-010) (Shariat, Hashemi, & Mohammadi, 2014). Since those recommendations were widely accepted, their underlying specifications were set as a standard in the industry, and so the first of its kind (Brown & Fallon, 2002). Although the AICC already tried to form a standard for e-learning content, the U.S Department of Defensive came up with the Advanced Distributed Learning Initiative[3] (ADL) in 1997 with the goal on modernizing the delivery of training to the forces (Brown & Fallon, 2002). Nevertheless it was not restricted to this purpose. In 1999 the first version of the e-learning specification was released. The ADL named it Shareable Content Object Reference Model[4] (SCORM) (Brown & Fallon, 2002).

Additionally three other organizations participated in creating specifications for e-learning at about the same period of time. The IMS [5] Global Learning Consortium which was founded in 1997 in the United States by organizations from the academic and business area and focuses mainly on metadata (Bakhouyi, Dehbi, Lti, & Hajoui, 2017). Europe's ARIADNE (Alliance of Remote Instructional Authoring Distribution Networks for Europe) can be seen as a complement to IMS. It was started by the European Commission's telematics for education and training program in 1996 and its top priority is also metadata (Ternier et al., 2009). The IEEE LTSC[6] (Learning Technology Standards Committee) develops agreed standards which is the result of independent evaluation made by individual experts instead of political or economic entities (Bakhouyi et al., 2017).

2.1.2 Learning Management Systems

The previous section served as an introduction into the history of e-learning. This section will deal with the idea of a Learning Management System

[2]HTML 4.01 Specification https://www.w3.org/TR/html4/, accessed 2017-11-16
[3]ADL http://www.adlnet.org/, accessed 2017-11-17
[4]SCORM https://scorm.com/, accessed 2017-11-17
[5]IMS https://www.imsglobal.org/, accessed 2017-11-26
[6]IEEE LTSC https://www.imsglobal.org/, accessed 2017-11-26

(LMS). In e-learning many different fields are combined and applied (Bowles, 2004). The author explains this by defining the term itself and reveals how each field participates to the process. Figure 2.1 shows the interaction.

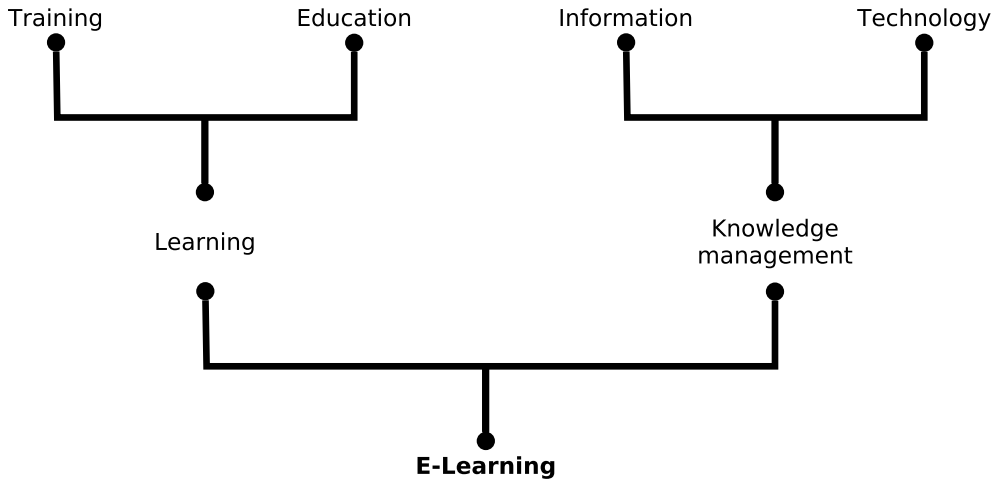

Figure 2.1: Interaction of different fields in e-learning. Image taken from Bowles (2004).

A learning management system (LMS) has to cover this and therefore also provide a flawless integration of all parts for the user. Based on the "Encyclopedia of the Sciences of Learning" learning management systems integrate interactive learning environments and administration and facilitate customized online instructional materials (Ifenthaler, 2012). Compared to learning content management systems (LCMSs) LMSs usually do not intend to create learning content which is the main difference. The author peruses the definition by stating that the LMS involves web-based software connected to a database, while the database contains different information types. By the use of widespread means including drag-and-drop individual courses can be made. The administration of all roles (instructors, students, courses) as well as the content is centralized and done automatically inside the system. (Ifenthaler, 2012)

As by the definition of Ifenthaler (2012) a learning management system has to fulfill a certain functionality. Figure 2.2 depicts the working principle of a LMS.

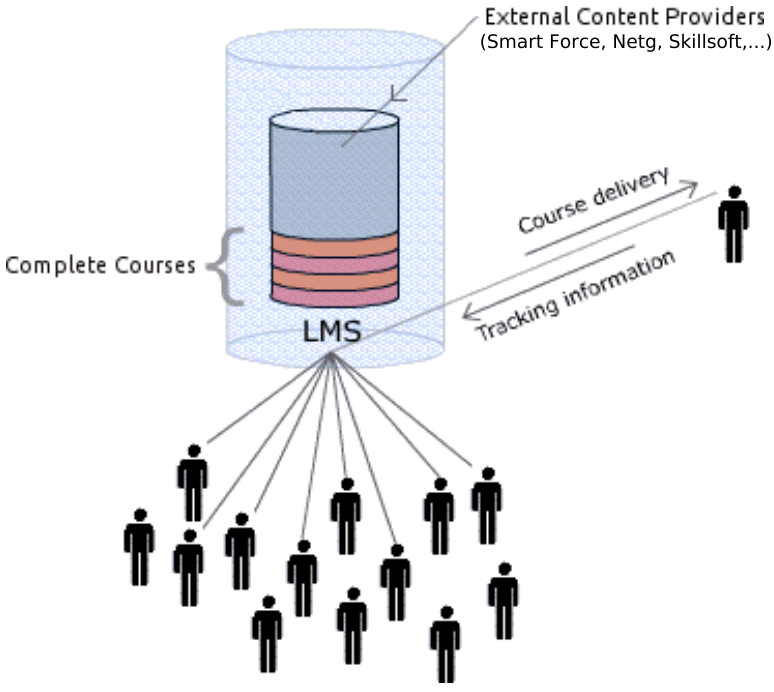

Figure 2.2: Working principle of a Learning Management System. Image taken and adapted from Nichani (2001).

2.1.3 Moodle

As one widely used representative of learning management systems Moodle (Modular Object-Oriented Dynamic Learning Environment)[7] is taken. It is a very famous and outstanding example among LMSs. To be exact it is an learning content management system which is a combination of a LMS and a content management system Nichani (2001). In Kumar, Gankotiya, and Dutta (2011) a comparison to other competitors in the field can be found. Compared to most other LMSs, which provide a tool-centered approach - this means that it gives a list of tools as interface, Moodle's focus on the other hand is learning-centered (Section 2.1.3.2) - it builds tools into an interface so the purpose of learning itself is emphasized (Cole & Foster, 2007). This is due to the background of its initiator which was in education

[7]Moodle https://moodle.org/, accessed 2017-11-26

(Cole & Foster, 2007).
The Moodle system is used as a main component in the lectures which the prototype in this thesis is built for and by which it should be extended. Hence, this section gives a short overview of its functional principle. Due to its large range of features the following explanation will only consider the general parts.

2.1.3.1 Structure

According to the paper of Kumar et al. (2011) Moodle consists of six different types of modules. Those are:

1. Communication modules
2. Productivity modules
3. Student involvement modules
4. Administration modules
5. Course delivery modules
6. Curriculum design modules

The following summary is based on Kumar et al. (2011).

(1) The communication modules are fundamental parts for additional communication functions which include file exchange, discussion forums and real time chat.

(2) Productivity modules are used to provide the user search functionality, a help and a calendar feature as well as tools for progress and review.

(3) Student involvement modules contain modules for workshop and group work. Additionally a module for student portfolio and self-assessment is included.

(4) Access to the administration modules means access to all other modules. Therefore these modules have to be used with care. This part of the LMS is responsible for authentication, authorization for users and to courses as well as the integration of registration.

(5) The use of the course delivery modules is normally restricted to teachers and administrative stuff. Submodules belonging to this are a helpdesk

module, tools for online grading, a course management module, a module for tracking students. Also testing modules are included in this group.

(6) In the last category customization modules and course templates can be found. Curriculum design modules are the parts that come along with the least chance of being attacked, since changes in the curriculum design can be easily identified compared to other modules.

2.1.3.2 Concept of Moodle

So far the basics of a learning management systems were covered. Also, the main components of Moodel were introduced. This section will target on its functionality and gives an overview on how Moodle works.

Learning Centered Approach

Moodle's focus is learning-centered (Cole & Foster, 2007). This differentiates it from most other learning management systems which are tool-centered (Cole & Foster, 2007). The basic idea of Moodle can be found in the term *social constructivism* (Cole & Foster, 2007). It describes a theory of knowledge which may answer best the question on how people learn together despite the fashion knowledge is transfered (Ifenthaler, 2012). Applied to Moodle construction of knowledge is a social process in which the sense of shared artifacts and symbols is negotiated by learners in order to gain a mutual understanding (Cole & Foster, 2007). Therefore Moodle provides tools to debate and share artifacts for which its *five key principles* serve as foundation (Moodle, 2010). Based on these facts, everyone that interacts with the system is at the same time a potential teacher and learner. Furthermore as soon as content for learning is created and expressed learning takes place. Another key is that one can also learn while studying others. In many situations knowing the background of learners helps to provide a better way of information transfer. Lastly, the learning environment plays an important role in order to establish a good learning process and should be flexible and adaptable to supply the user with quick responses.

Course Concept

According to Cole and Foster (2007) Moodle's functionality is based on courses. Compared to other systems which support only one format Moodle allows to use several alternatives. In case the format does not fit it is also possible to switch. The following formats are offered (Cole & Foster, 2007):

- LAMS[8] (Learning Active Management System) format: An open source learning management system which provides teachers a flash based authoring environment.

- SCORM format: The Shareable Object Reference Model was already introduced in section 2.1.1. It is a standard for wrapping content. Packages in this format contain JavaScript activities and its appropriate content.

- Social format: The foundation for this format is an individual forum for the entire course which is usually the case if it is less formal.

- Topics format: By using this format Moodle will create a topic for every section. It makes sense to use this format for concept-oriented courses.

- Weekly format and CSS/no tables: Compared to the previous format this is applied if terms have to be complied. Visual information of the progress is provided.

In figure 2.3 a simple course page is shown. This page contains three areas. On the left hand side tool blocks can be found. The center block lists the course content and activities. The area on the right hand side shows news related to the subjects that are taught or learned.

[8]LAMS https://www.lamsinternational.com/, accessed 2018-01-02

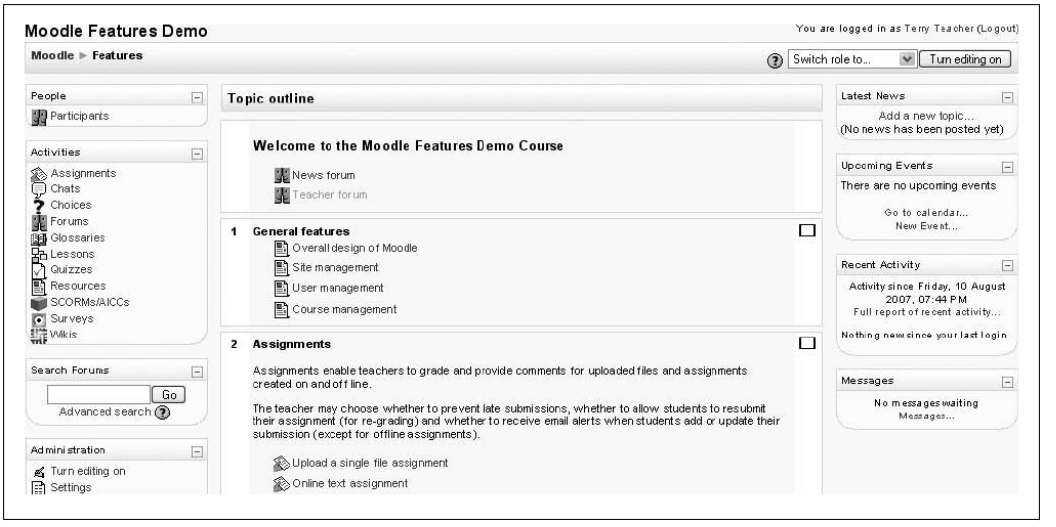

Figure 2.3: A simple course page from the Moodle learning system. Image taken from Cole and Foster (2007).

2.1.4 Learning Styles

> *"Learning styles are considered by many to be one factor of success in higher education."* (Romanelli, Bird, & Ryan, 2009)

In the article from Sternberg and Grigorenko (1997) learning styles are seen as a subset of cognitive styles which can be mainly put into cognition centered, personality centered, and activity centered. The author classifies the activity-centered cognitive style as the only real style for learning. In Kolb (1984) one can find one of the most popular theories in education - The Learning Style Inventory. Based on this learning can be split into four phases: concrete experience (CE), reflective observation (RO), abstract conceptualization (AC), and active experimentation (AE) which forms the learning cycle. Based on McLeod (2013) the phases of the learning cycle are explained as follows:

- **Concrete Experience**: This is the starting point of Kolb's cycle. It takes place when a new situation or experience occurred. Active contribution, either as individual or as a group is the primary factor.
- **Reflective Observation**: In this stage the previously encountered experience is reviewed. Usually questions are asked and communication

happens.

- **Abstract Conceptualization**: New ideas or existing notions are converted or altered. Additionally, analyses are conducted and conclusions generated.
- **Active Experimentation**: Conclusions are applied in future situations in order to gain new experiences.

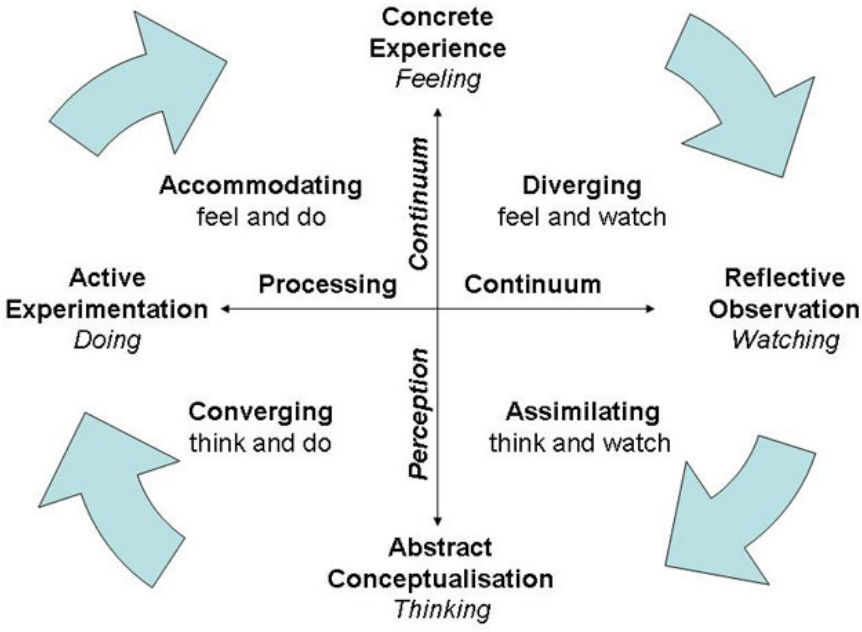

Figure 2.4: Learning Style Cycle. Image taken from McLeod (2013).

According to Kolb's opinion it is not possible to carry out two actions located oppositely in the cycle (e.g. Concrete Experience and Abstract Conceptualization) (McLeod, 2013). Instead learning styles emerge when combining axes next to each other. This leads to new definitions of learning styles. In figure 2.4 the circumstances can be seen. Based on the summary of McLeod (2013) those are described as follows:

- **Diverging**: This style is created when Concrete Experience and Reflective Observation in combined. It tackles feeling and watching.

People responding to those have the ability to look at things from different points of view and prefer watching instead of doing in order to gain information. Emotions and imagination play a big role and an interest in other people can be seen. Therefore, those people prefer teamworks.

- **Assimilating**: People who favor Reflective Observation (watching) and Abstract Conceptualization (thinking) use the learning style Assimilation. They prefer to deal with concepts and ideas instead of working with other people. Furthermore, they perform well when its about widespread information and tend to work better with logical theories, which can be reasoned out, rather than practicals.
- **Converging**: This learning style connects Abstract Conceptualization and Active Experimentation. People who fall into this category learn by tackling technical problems to find solutions. Social interaction with other people plays a minor part. Such learners like experimenting with ideas. They are good in converting theoretical concepts to practical uses while they can "think and do".
- **Accommodating**: Active Conceptualization and Concrete Experience are combined to get a "feel and do" learning style. Sense is more important than logic which results in trust and reacting based on the gut feeling. Those learners apply analyses from others and like practical approaches. It is the predominant learning style.

Furthermore, Kolb's model serves as a foundation for other models (Huang, Lin, & Huang, 2012) such as the Learning Style Questionnaire by Mumford and Honey (1992) which is used to identify the type of a learner by investigating general behavioral tendencies instead of inquiring straightly the learning behavior (Huang et al., 2012).

However, most of the models share same or similar concepts and basics that belong to learning (Huang et al., 2012). Much more important is the fact that every learner needs to be exposed to all of this styles with respect to their individual sensitivity (McLeod, 2013). The applied learning style does also depend on the topics that are taught, the age and the level of education (Othman & Amiruddin, 2010).

2.2 Concepts in Information Retrieval

In this section the main concepts of information retrieval, which are taught in the course, are discussed. The following discussion deals with the background that is implemented in the prototype. It starts with some important definitions which are then used to explain the architecture of searching that is shown in figure 2.5. This is followed by preprocessing steps. The last part of the concepts-section addresses retrieval models.

2.2.1 Definitions

In order to explain concepts in information retrieval, some terms in that field need to be defined. This will start with information retrieval itself.

2.2.1.1 Information Retrieval (IR)

There are many definitions of this term, but all of them deliver a similar message.

> *"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually test) that satisfies an information need from within large collections (usually stored on computers."*(Manning, Raghavan, & Schütze, 2008)

> *"Information retrieval deals with the representation, storage, organization of, and access to informations items such as documents, Web pages, online catalogs, structured and semi-structured records, multimedia objects."*(Baeza-Yates & Ribeiro-Neto, 2011)

Information retrieval is the process of finding existing information instead of new structures in data. For clarification another strategy is introduced. This is called data retrieval and should help to define the term IR. Data retrieval searches for patterns and strings whereas IR searches for content and information and the query statement can be fuzzy or complex. In data retrieval queries have to be defined exactly and results are equally relevant.

This does not hold for IR. The result set usually has a ranked order. To finalize the comparison it is worth to mention that information retrieval is error tolerant, whereas the process of data retrieval is not. Even though these strategies are different approaches for retrieval systems, they are often used together. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.2 IR Model

Based on the definition from Baeza-Yates and Ribeiro-Neto (2011) an IR-Model is a quadruple $[\mathbf{D}, \mathbf{Q}, F, R(q_i, d_j)]$. $\mathbf{D}$ represents a set that contains logical representations of documents in a collection. $\mathbf{Q}$ stands for the set of logical representations of user needs that are also called queries. In order to map documents, queries and their relations, a framework is needed. This is represented by F. The last part of the quadruple is a ranking function $R(q_i, d_j)$. This function maps the similarity of a document $d_j \in \mathbf{D}$ and a query $q_i \in \mathbf{Q}$ to a real number. This mapping applied to all documents results in a ranking with respect to a query q_i.

2.2.1.3 Document Collection

A document collection is a collection that contains all documents to be indexed. In section 2.2.2 properties of such a collections will be listed.

2.2.1.4 Reference Collection

A reference collection is a predefined and well selected document collection. It is used to evaluate retrieval performance. Famous reference collections are the TREC [9] or the Cranfield Collection[10]. (Manning et al., 2008; Baeza-Yates & Ribeiro-Neto, 2011)

[9] TREC https://trec.nist.gov/, accessed 2017-7-2

[10] Cranfield-Collection http://ir.dcs.gla.ac.uk/resources/test_collections/cran/, accessed 2017-7-2

2.2.1.5 Index Terms - Keywords

Index terms or keywords are the parts that contain the information of a document an describe it. Basically any word can be used as a keyword, but usually those that are relevant to a topic are selected. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.6 Vocabulary

A vocabulary is the set of words in the document collection that are indexed. So if k_i is considered as a keyword and t represents the number of terms to be indexed, then the vocabulary $V = \{k_1, ..., k_t\}$. Since the vocabulary is a set, words are unique. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.7 Document Representation

The representation of a document is based on its term co-occurrences that form patterns. In other words a document can be described by the occurrence or the frequency of occurrences of its terms. This also refers to the "Bag of Word" - model of a document. In order to index this patterns for all documents a so-called term-document-matrix is used. (Manning et al., 2008; Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.8 Query Representation

The representation of the query has the same structure as the one for the document. Since the query can be seen as a single document, the resulting representation will be a term vector. (Manning et al., 2008)

2.2.1.9 Term-Document - Matrix

This matrix C with the format MxN is a simple data structure for indexing documents. Each row represents a term k_i, each column a document d_j. If a term appears in a document, the cell contains an entry $z_{i,j}$. An entry

can model the occurrence of a term, its frequency or a weight, which leads to term weighting (section 2.2.1.10). All of them indicate appearance or significance of a term in a document. (Manning et al., 2008) In the following illustration the structure of the matrix is shown.

$$C = \begin{array}{c} \\ d_1 \\ \vdots \\ d_N \end{array} \begin{array}{c} \begin{array}{cccc} k_1 & \dots & k_M \end{array} \\ \begin{bmatrix} z_{1,1} & \dots & z_{1,M} \\ \vdots & \ddots & \vdots \\ z_{N,1} & \dots & z_{N,M} \end{bmatrix} \end{array}$$

2.2.1.10 Term Weighting

The concept of term weighting is used to assign an importance to a term in a document. In order to do so a weight $w_{i,j} > 0$ is associated with each term k_i that contains in a document d_j. For a term that is not in the document the weight equals zero. By the use of weights a ranking scheme can be applied to each document in a collection with respect to a query. Weighting scheme also depends on the IR-model. Further explanation on the use of term weighting can be found in section 2.2.4.2. (Manning & Schütze, 1999)

2.2.1.11 Inverted Index

Inverted index is a data structure for indexing. It is word-oriented and solves the problem that comes along with term-document matrices. These matrices are often very sparse since not all words appear in all documents. Due to this they slow down the searching task and also waste memory space. To face this an inverted index contains for each word k_i a document list, that indicates in which documents k_i occurs. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.12 Levenshtein Distance

The Levenshtein Distance is a measure that provides information on how many edit operations (insert, delete or replace) are necessary to convert one string into an other. It is an integer that is used to state the similarity of two strings. (Manning et al., 2008)

2.2.1.13 Query Expansion

Query Expansion is an approach to extend a user query with terms in order to provide better results. This can be achieved by the feedback information of the user or automatically. When the user is involved in the process relevant documents are indicated and according to these explicit help is provided. The implicit query expansion is based on two strategies. On one hand local analysis and on the other hand global analysis. Local analysis uses the information from the top ranked documents in a result set while global analysis makes use from external resources in order to attain information for query improvement. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.14 Fuzzy Set Model

The Fuzzy Set Model is an alternative model based on the set theory. This approach provides a framework that considers the concept of term dependencies. Usually the correspondence of a query and a document is an approximation or vague. To tackle this vagueness the fuzzy model assigns each query term a fuzzy set and each document obtains a degree of membership in the set. The intension of this model is to address the representation of classes without clear separations. The degree of membership is a value in the interval $[0, 1]$. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.15 HITS

HITS stands for hyperlink-induced topic search and is a ranking scheme (Manning et al., 2008). The main entities are hubs and authorities. Hubs

h_i are pages that refer to many other pages containing relevant content. Authorities a_i are pages that contain relevant content.

The algorithm works in an iterative manner and contains three steps. Starting with $a_i = h_i = 1$:

1. Calculate new values for authorities a'_{p_i} using equation 2.1
2. Calculate new values for hubs h'_{p_j} using equation 2.2
3. Normalize values using equation 2.3 and 2.4

h_{p_i} and a_{p_i} represent hub and authority values for page p_i.

$$a'_{p_i} = \sum h_{p_j} \tag{2.1}$$

$$h'_{p_i} = \sum a_{p_j} \tag{2.2}$$

$$\sum (a'_{p_i})^2 = 1 \tag{2.3}$$

$$\sum (h'_{p_i})^2 = 1 \tag{2.4}$$

The iteration will be repeated k cycles until values start to converge. HITS algorithm assumes that good hubs point to many authorities and authorities can be reached by many hubs. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.1.16 Page Rank

PageRank is a ranking scheme used by Google. The scheme computes the popularity of a link. It models how a user surfs the web. The user follows hyperlinks and jumps to other pages. PageRank calculates the probability of visiting a page based on the structure of the links. The probability is influenced by the number of inbound links. A high number results in a high popularity of a page.

Equation 2.5 shows the algorithm where $PR(a)$ states the PageRank for page a:

$$PR(a) = \frac{q}{T} + (1 - q) \sum_{i=1}^{n} \frac{PR(p_i)}{L(p_i)} \tag{2.5}$$

$L(p)$ is the number of outgoing links of page p, $p_1 \dots p_n$ represent pages which point to page a, q shows the probability a user changes to a random page and $1 - q$ is the probability that a user will click a link at the current page. Additionally T states the number of all pages and q is a parameter that can be adjusted by the system and is usually set to $0,15$. Initially all pages have a PageRank PR of:

$$PR(p_i) = \frac{1}{T} \tag{2.6}$$

Like in the HITS algorithm after an iteration a normalization has to be done. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.2 Architecture for Searching

Figure 2.5 gives an overview of the steps that are applied in the process of information retrieval. The following subsection will explain all important parts of the process. These steps can be applied for both information retrieval and data retrieval. Following explanations of the individual steps are based on the book of Baeza-Yates and Ribeiro-Neto (2011).

2.2.2.1 Document Collection

A document collection is the set of all documents. These documents represent the original information content and can exist in different formats or contain different formats. Either way, for further processing text operations have to be applied in order to prepare it for the process of information retrieval. This can be done by human interaction, in an automated manner or in a combined action of both. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.2.2 Text Operation

Text operations can be split into a preprocessing and a post processing part. In the first part documents are unified by format converting and cleaned. It is a huge step in the process of IR because it does not only

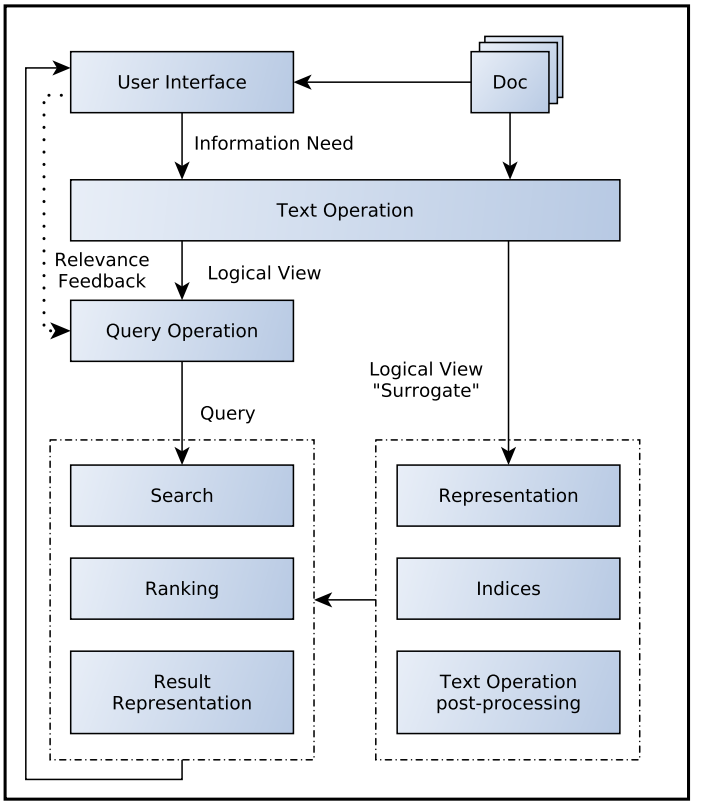

Figure 2.5: Conceptual Architecture for Searching. Image adapted from Baeza-Yates and Ribeiro-Neto (2011).

involve the conversation from different formats to a single one, but also operations like tokenization, the extraction of words and structure and natural language processing (NLP). Tokenization is the process of splitting a text stream into tokens, typically words. NLP deals with operations on natural language like removing stop words, finding noun groups (part-of-speech tagging (Manning & Schütze, 1999)) or stemming (Sun, Liu, Hu, & Zhu, 2014). Post processing deals with the presentation of the search result. Section 2.2.3 covers text operations and explains each single step. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.2.3 Internal Document Representation

The internal document representation can also be seen as the logical view of a document. This includes the representation itself which can contain meta data, parts of a text or the full text and the structure of the text. By the use of data structures the retrieval process can be expedited. In this context the inverted index comes into play which is a data structure for looking up documents through words. This representation also adds support for retrieval features that are not maintained by an IR-model. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.2.4 Information Need

This part includes the user interface, text operation and query operation. Information need that is entered in the user interface needs to be converted into a query. Since queries and documents have to have the same format queries need to be standardized. Therefore the same text operations as for documents have to be applied. After cleaning and adapting, the user query needs to be transformed into a system query. This is the query operation and depends on the IR-model. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.2.5 Search Operation

Search operation is the step in which the query "faces" the documents. The format of the system query depends on the query language and the system features. Ranking and retrieval depend on the retrieval model and the representation. For the result representation any reasonable format can be used. This could be for instance text, metadata (section 2.2.3) or even a visualization of the result. Finally, the chain of retrieval process contains a relevance feedback which provides help based on the results of a previous search request in order to improve the user query and to create a new one. This can happen manually, semiautomated or automated. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.2.6 Query Languages

This small section provides additional information regarding the creation of queries. A query is used to specify the information need and it should be possible to define a granularity. The query language is defined by the IR-model and the internal document representation. It is also influenced by the user's input, that may differ in type, as well as the output format. In information retrieval keyword-based querying is used. That means the user can define its information need by a number of words, which correspond to relevant documents. Based on the IR-model a query can be assembled more or less complex, starting with queries containing single words to combinations of words. Some models allow to create boolean queries which means that keywords are combined by boolean operators. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.3 Preprocessing

Depending on the source of the documents, a retrieval system has to deal with various different formats. This section addresses the steps that are necessary to convert a document into its representation. The basic process includes document and text processing, document representation and indexing which is used to speed up the retrieval process. In Figure 2.6 the fundamental workflow is shown and will be explained hereafter whereas the explanation is based on the book of Baeza-Yates and Ribeiro-Neto (2011).

2.2.3.1 Document

Based on the definition of Baeza-Yates and Ribeiro-Neto (2011) a document represents an individual entity of information that can be for instance a book, an email or an article. Characteristic for such a unit is that it follows a syntax and has a structure. It also contains semantics which are connected to the author. Additionally a presentation style can be included. (Baeza-Yates & Ribeiro-Neto, 2011)

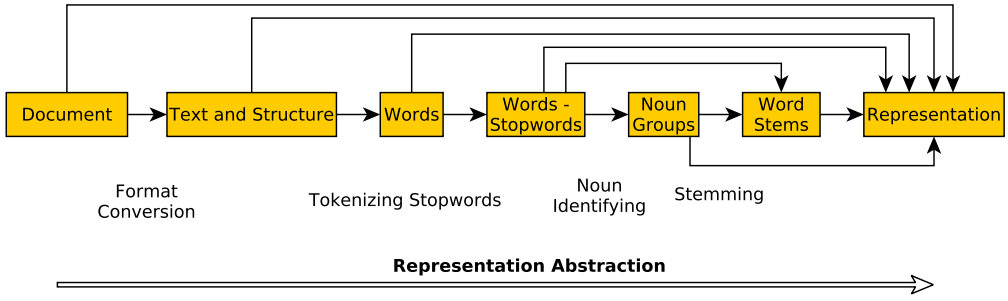

Figure 2.6: Basic steps to build a model in IR. Image adapted from Baeza-Yates and Ribeiro-Neto (2011).

2.2.3.2 Text and Structure

Appearance of a document is defined by a syntax, semantic and a presentations style as mentioned previously. The document style is responsible for the visualization in digital or printed form. Apart from this, metadata can be found in documents that is related to its content. Metadata is extra information that describes the content, hence it can be seen as data about the data. It can be distinguished between descriptive and semantic metadata. While the first one is external and explains the sense of the document and its creation, the latter one characterizes the topic inside the content of the document like how parts are put together to form the document. Metadata standards are MARC[11], Dublin Core[12] and Learning Object Metadata (LOM)[13] that is used in e-learning (Pansanato & Fortes, 2005). (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.3.3 Format Conversion

This is the step in which documents that exist in different file formats are converted into a internal representation. The representation considers text and structure. For that process open source solutions might be used, but proprietary formats can cause problems. Documents can be internally

[11]MARC http://www.loc.gov/marc/, accessed 2017-8-26
[12]Dublin Core http://dublincore.org/, accessed 2017-8-26
[13]LOM https://ieee-sa.imeetcentral.com/ltsc/, accessed 2017-8-26

represented in a standard like the standard object interchange format or a markup language like HTML or XML. (Manning et al., 2008)

2.2.3.4 Tokenization

In this process a text stream is split up into tokens. Tokens are elements that contain a meaning like words, phrases or symbols. They are separated by delimiters, usually whitespace or punctuation characters, which denote an end of a sentence or another important section. While for most of the time these separators seem to be a good choice, there may be situations in which they do not perform very well. A simple counterexample would be the URL of a web-page that contains dots in it. Other problems that may appear in the process are those that rely on word representations. Since the tokenization process also removes useless characters, abbreviations like "E-Mail" can lead to issues. Also numbers without a context cannot be used or need to be treated separately. (Manning et al., 2008)

2.2.3.5 Words - Stopwords

In natural text the distribution of words is very biased (Manning & Schütze, 1999). A few words that appear very often can cover up to 50% of the text (Baeza-Yates & Ribeiro-Neto, 2011). Considering that, words that appear very frequently can be dropped. These words are called stop words and usually do not contain any relevant information (Baeza-Yates & Ribeiro-Neto, 2011). This was already suggested by Luhn (1958b) who revealed that the frequency of words can be used to measure the significance of words. By removing them the size of the index can be reduced and speed up the retrieval process. Such words are mostly articles, prepositions and conjunctions. To eliminate stop words a histogram can be used that gives an overview of the word distribution in a document collection. Another way to identify potentially useless parts is to apply lexical analysis that makes use of a predefined stop-word-list or a thesaurus lookup. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.3.6 Noun Groups

Words that contribute most information to texts are nouns and noun groups. Therefore this is the kind of words that is used as index terms. Extracting of those important pieces is done via part-of-speech tagging, thesaurus lookup and wordnet lookup which is all part of natural language processing (NLP) (Manning et al., 2008). Most of noun groups contain additionally an article, a possessive, a demonstrative, a quantifier, a numeral or a question word. Plenty of them also involve modifiers such as adjectives or a secondary noun (Lau & Lutz, 1973). (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.3.7 Stemming

In documents same words appear in different variants. These variations are also known as inflection forms. Depending on the type of the word variants include plurals, gerund forms or suffixes related to a tense form of a verb. To join different word variants into a common base two approaches can be used. Both of them stemming and lemmatization are part of natural language processing. Stemming cuts off derivational affixes to convert a word to its stem by applying rules. A very well known algorithm is the Porter-stemmer (Baeza-Yates & Ribeiro-Neto, 2011). Lemmatization on the other hand uses an approach that is based on a vocabulary and a morphological analysis of words to retrieve its roots which are called the lemma (Manning et al., 2008). The result after the transformations process is again a reduction of the index size. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.3.8 Document Representation

The question of how a document should be represented and which parts should be indexed is crucial. It depends on factors like the used model and the user needs. A document can be described by a few keywords that represent the main parts which sum up the topic or by the full text. For partial indexing typically an expert selects the index terms. It is also possible to automate the selecting process by using intelligent systems that apply tools from natural language processing. (Baeza-Yates & Ribeiro-Neto, 2011)

2.2.4 Retrieval Model Strategies

In this section three fundamental and widely used models in information retrieval will be discussed, namely the Boolean Model, the Vector Space Model and the Probabilistic Model (Baeza-Yates & Ribeiro-Neto, 2011). Basically a model is a method that is used to calculate the relevance of a query to a document (Pannu, James, & Bird, 2014). The mentioned models are the fundamentals in terms of retrieval strategies. Pannu et al. (2014) compares the basic models regarding their advantages and disadvantages. There are more advanced methods which optimize or extend the existing ones. Also combinations of the basic models are used. A good example for this may be Apache Lucene[14] which makes use of both the boolean approach and the vector space model. Other strategies also include approaches from the field of machine learning like neural networks (Baeza-Yates & Ribeiro-Neto, 2011) or genetic algorithms (Cerulo & Canfora, 2004).

2.2.4.1 Boolean Model

One of the simplest models is the boolean model. According to Baeza-Yates and Ribeiro-Neto (2011) it is based on the set theory and allows queries that connect keywords with binary operators (AND, OR, NOT). Given a document d_j and a query q the similarity is then given by:

$$sim(d_j, q) = \begin{cases} 1 & \text{if } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall_{k_i},\ g_i(\vec{d_j}) = g_i(\vec{q}_{cc})) \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

$\vec{q}_{dnf}$ is a query q in disjunctive normal form and $\vec{q}_{cc}$ represents the individual parts of $\vec{q}_{dnf}$. Based on the definition in equation 2.7 weights $w_{i,j}$ are binary. There is only a match between a query q and a document d_j if $sim(d_j, q) = 1$. Due to the simple matching principle no ranking in the result set exists. Each document is either relevant or not relevant. (Baeza-Yates & Ribeiro-Neto, 2011)

[14] Lucene https://lucene.apache.org/core/2_9_4/scoring.html, accessed 2017-11-10

2.2.4.2 Vector Space Model

This model has a long history. Already Salton, Wong, and Yang (1975) showed a model that was used for automatic indexing. The base of this model is the vector space in which both documents and the query are mapped in order to find a similarity. In this model word frequencies are taken into account. It overcomes the problem of the boolean model which restricts the result set to only full matches. The vector space model uses non-binary weights in its retrieval process and allows partial hits. The similarity calculation involves all documents in the collection that face the query. After this process the similarities are sorted. This leads to a ranked order. (Baeza-Yates & Ribeiro-Neto, 2011)

Let the following four documents be the document collection.

- Document d1: *word1.*
- Document d2: *word1 word2 word3.*
- Document d3: *word1 word1 word3.*
- Document d4: *word1 word2 word2 word3 word3 word3.*

In figure 2.7 one can see its mapping in the vector space.

In contrast to the boolean model the vector space model considers the frequency of a word in a document and also the number of documents in which a word appears in the overall collection. Instead of just taking the raw word counts a weighting scheme is used. It is called TF-IDF and stands for term-frequency - inverse document frequency. The term-frequency refers to the importance of a term in a document while the second part of the scheme indicates the importance of a term in the whole document collection. Following equations will show how it is obtained. (Baeza-Yates & Ribeiro-Neto, 2011)

$$tfidf = tf * idf \tag{2.8}$$

$tfidf$ corresponds to the weight w of a term. Whereas the term-frequency weight tf is calculated by using the scheme in equation 2.9. $f_{i,j}$ is the word frequency of the word i in the document j. (Manning et al., 2008)

$$tf_{i,j} = \begin{cases} 1 + \log_2 f_{i,j} & \text{if } f_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

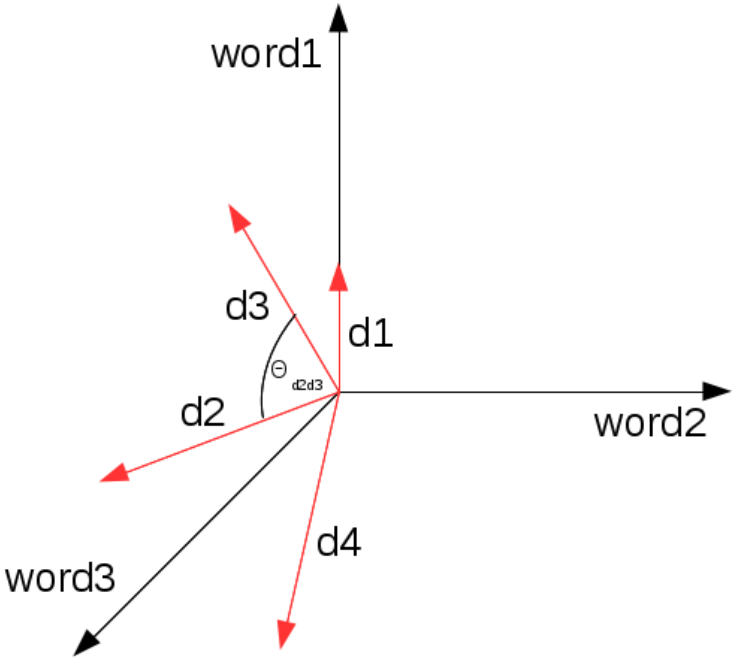

Figure 2.7: A simple Vector Space Model with four documents mapped in three dimensions.

The term-frequency gives information about the importance of a term within a document. Normally is done by the use of a function like the logarithm, like in equation 2.9, or the square root. This is due to a higher importance of a term which is the result of a higher occurrence, but with increasing frequency its importance does not grow that much anymore. (Manning & Schütze, 1999)

For the inverse-document-frequency weight idf the scheme in equation 2.10 is used. N states the number of total documents and n_i the number of documents containing the word i. (Baeza-Yates & Ribeiro-Neto, 2011)

$$idf_i = \log_2 \frac{N}{n_i} \tag{2.10}$$

There exist also other variations of this weighting (Manning et al., 2008), but the versions used in this work are commonly used and proven to perform well as pointed out by Manning and Schütze (1999).

As ranking function the cosine similarity is used. It calculates the cosine of two vectors which is in the interval of $[0, 1]$ (Manning & Schütze, 1999). A score of 0 means there is no similarity while a score of 1 indicates total similarity. Usually this equation is used to calculate the angle (Li & Han, 2013), but since the cosine is a monotonically decreasing function of the angle it changes proportionate to the angle and can therefore be used instead which reduces computation time. The following equation states the similarity of a query q and a document d. (Manning & Schütze, 1999)

$$sim(q, d) = \cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \sqrt{\sum_{i=1}^{n} d_i^2}} \tag{2.11}$$

$\vec{q}$ is the query vector and $\vec{d}$ represents a document as vector of the same length n in the document collection. Both contain the TF-IDF weights.

2.2.4.3 Probabilistic Model

A classic probabilistic model was already proposed by Manor and Kuhns in 1960 (Singhal, 2001). The model that was used in this thesis is called the binary independence model (BIM) and was introduced in 1976 (Baeza-Yates & Ribeiro-Neto, 2011). Like the vector space model, BIM also considers word appearances in a document collection, but not appearances in the document (Manning et al., 2008). Further developments of this model like the Okapi BM25, an advanced probabilistic model, also consider word frequencies in a document (Manning et al., 2008).

In BIM both documents and queries are seen as incidence vectors that show only the presence or absence of a term and therefore contain binary weights $w = \{1, 0\}$ that determine the presence or absence of a term. The expression "independent" states that terms are independent of each other in documents, which is usually not the case, but works in practice most of the time. Based on this independence assumption Naive Bayes models work. (Manning et al., 2008) (Baeza-Yates & Ribeiro-Neto, 2011)

The following parts of section 2.2.4.3 are based on the book of Baeza-Yates and Ribeiro-Neto (2011) and will summarize concepts that were used in the prototype of this thesis. The fundamental of BIM is the *Bayes' rule*, which is stated in equation 2.12. The following shows the derivation of this ranking model.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.12}$$

As an entry point for the **derivation of the classic probabilistic model** the similarity between a document d_j and a query q is used (Baeza-Yates & Ribeiro-Neto, 2011). This is given by:

$$sim(d_j, q) = \frac{P(R|\vec{d}_j)}{P(\overline{R}|\vec{d}_j)} \tag{2.13}$$

Using *Bayes' rule* we get

$$sim(d_j, q) = \frac{P(\vec{d}_j|R, q)P(R, q)}{P(\vec{d}_j|\overline{R}, q)P(\overline{R}, q)} \tag{2.14}$$

- $P(\vec{d}_j|R, q)$ indicates the probability of a randomly selected d_j from the set of relevant documents R
- $P(R, q)$ stands for the probability of selecting a relevant document from the whole collection
- Correspondingly $P(\vec{d}_j|\overline{R}, q)$ and $P(\overline{R}, q)$ are the complementary parts

By neglecting parts that are constant for all documents [$P(R, q)$,$P(\overline{R}, q)$] the formula simplifies to

$$sim(d_j, q) \sim \frac{P(\vec{d}_j|R, q)}{P(\vec{d}_j|\overline{R}, q)} \tag{2.15}$$

Considering the independence of index terms and binary weights $w_{i,j}$ lead to

$$sim(d_j, q) \sim \frac{(\prod_{k_i|w_{i,j}=1} P(k_i|R,q)) \times (\prod_{k_i|w_{i,j}=0} P(\overline{k_i}|R,q))}{(\prod_{k_i|w_{i,j}=1} P(k_i|\overline{R},q)) \times (\prod_{k_i|w_{i,j}=0} P(\overline{k_i}|\overline{R},q))} \tag{2.16}$$

- $P(k_i|R,q)$ is the probability that a term k_i occurs in a randomly selected document from the set R
- $P(\overline{k_i}|R,q)$ indicates a the probability that term k_i is not present in a randomly selected document from the set R
- Correspondingly terms $P(k_i|\overline{R},q)$ and $P(\overline{k_i}|\overline{R},q)$ are the complementary parts

by taking into account that $P(k_i|R,q) + P(\overline{k_i}|R,q) = 1$, neglecting constant factors for documents for the same query and applying the logarithm we obtain a key expression for calculating the ranking in the probabilistic model.

$$sim(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log \frac{P(k_i|R,q)}{1 - P(k_i|R,q)} + \log \frac{1 - P(k_i|\overline{R},q)}{P(k_i|\overline{R},q)} \tag{2.17}$$

Since there are no retrieved documents in the beginning an initial guess has to be made in order to have starting values for $P(k_i|R,q)$ and $P(k_i|\overline{R},q)$. This guess is based on the following assumptions.

- $P(k_i|R,q) = 0.5$
- $P(k_i|\overline{R},q) = \frac{n_i}{N}$

After applying these assumptions we get

$$sim(d_j, q) \sim \sum_{k_i \in q \wedge k_i \in d_j} \log \frac{N - n_i}{n_i} \tag{2.18}$$

N stands for all documents, n_i indicates documents containing term k_i. After the initial guess the procedure can be repeated to enhance the result. In order to do that a subset of the initially retrieved documents (e.g. first 20 top ranked entries) is used.

Note: Equation 2.18 represents a IDF-like ranking like which is used in the vector-space model.

- $P(k_i|R,q) = \frac{D_i+0.5}{D+1}$
- $P(k_i|\overline{R},q) = \frac{n_i-D_i+0.5}{N-D+1}$

D_i represents the subset of initially retrieved documents, D represents the total number of initially retrieved documents. The factors 0.5 and 1 are used to prevent problems for cases $D = 1$ and $D_i = 0$. It should be mentioned that the constant factor of 0.5 is not always the best choice even though it was used in the prototype of this thesis. Instead of this the ratio of $\frac{n_i}{N}$ can be taken as an alternative. (Baeza-Yates & Ribeiro-Neto, 2011)

2.3 Text statistics

In the following section important text statistics are explained. Usually natural text has semantics and follows a syntax. These statistics are used to describe the behavior of natural text. They can be applied to various languages and may also help to analyze documents in a way to improve the retrieval mechanism and get a better understanding of what is happening. (Baeza-Yates & Ribeiro-Neto, 2011)
Results of these statistics also play a role in linguistics and natural language processing (Altmann & Gerlach, 2016). An application could be the distinction of languages based on their word-length, mean and entropy (Kalimeri et al., 2014). Another field where statistics are used is communication engineering. Shannon (2001) showed by the use of entropy that any noisy communication channel can be used to transfer discrete information almost without errors. Entropy is also used to compress text. A well known example is the huffman coding (Huffman, 1952).

2.3.1 Single Letter Characteristics

This group refers to statistics of letters in a text. It includes information theory and the letter frequency. For the prototype in this thesis only the letter frequency is important.

2.3.1.1 Letter Frequency

The Letter Frequency shows a probability distribution of letters in a text. It depends on the language and can also be used to identify the language of a text. This is revealed by (Takcı & Soğukpınar, 2005) who applies the vector space model presented in 2.2.4.2 to letters in order to perform the distinction. The following definition was taken from Takcı and Soğukpınar (2005).

$$\vec{d}_{lf} = (lf_1, lf_2, ..., lf_n) \tag{2.19}$$

$\vec{d}_{lf}$ is the representation of a document. lf_i states the frequency of a letter i in a document d.

2.3.2 Single Word Characteristics

In contrast to single letter characteristics this section operates a "level" above since it entirely concerns words. The law stated in subsection 2.3.2.1 is the basis of all word statistics used in this thesis.

2.3.2.1 Zipf's Law

Zipf's Law (Zipf, 1949) reveals a relation between organizations and its rankings. Applied in text statistics it shows that if terms of a vocabulary in a document are sorted in decreasing order of their frequencies the inverse of the frequency match the ranks of the terms. (Baeza-Yates & Ribeiro-Neto, 2011)

$$f(r_i) \sim r_i^{-\alpha} \tag{2.20}$$

f is the frequency of the i-th ranked word and r_i its rank. α is an empirical constant with $\alpha \geq 1$. In the original version of Zipf's Law $\alpha = 1$, but for natural text it varies is lies somewhere in the range of $1 \leq \alpha \leq 2$ (Ferrer i Cancho, 2005). There is a relation between the frequency and the rank of a word which is given by a constant k (Manning & Schütze, 1999).

$$f * r = k \tag{2.21}$$

For clarification of Zipf's Law let the following be considered as an example. The word with the rank $r = 50$ should occur with three times of the frequency of the word at rank $r = 150$ (Manning & Schütze, 1999).

2.3.2.2 Word Length Distribution

The word length distribution is the foundation of the average word length L_{av} which is very much influenced by stopwords since they are usually articles or conjunctions - terms with a shorter word length (Manning et al., 2008) - and also depends on the language (Bochkarev, Shevlyakova, & Solovyev, 2012). It is given by:

$$L_{av} = \sum_{i}^{N} p_i l_i \tag{2.22}$$

where l_i indicates the length of the i-th word and N the number of distinct words. p_i is defined as the probability of the i-th word.

$$\sum_{i}^{N} p_i = 1 \tag{2.23}$$

The definition was taken and slightly adapted from Bochkarev et al. (2012)

2.3.2.3 Word Frequency

This statistic answers the question of how words are distributed in a text. It shows the probability of the occurrence of words (Allahverdyan, Deng, & Wang, 2013, 6).

$$\{f_r\}_{r=1}^{n},\ f_1 \geq \cdots \geq f_n,\ \sum_{r=1}^{n} f_r = 1 \tag{2.24}$$

Where n is the number of unique words and f_r represents the frequency (probability) of word r.

Languages in general show a similar behavior since its word distributions follow a power law (Piantadosi, 2014) that refers to Zipf's Law in section 2.3.2.1. That means a few words appear very often while many appear very seldom. The three most frequent words (for example stopwords) can make up to 12% of the total distribution. On the other side in a large corpus it is very likely that about 50% of words appear infrequently or even only once as stated in Manning and Schütze (1999). This phenomena is also known as *hapax legomena* (Greek for "read only once") (Baker, Hardie, & McEnery, 2006).

This information can be used in information retrieval. Luhn (1958a) proposed that words that occur very often and words that occur very infrequently should not be used for indexing.

As concluded by Zipf's law multiplying the frequency f_r of a word multiplied with the rank r leads to a constant also the multiplication of a word's rank r with its probability p_r does. The result is the constant $A \approx 0.1$.

$$p_r = \frac{f_r}{f} = \frac{A}{r} \tag{2.25}$$

f stands for the total number of word occurrences. This equation again can be rewritten to:

$$p_r * r = A \approx 0.1 \tag{2.26}$$

2.3.2.4 Rank Of Given Word Frequency

By the use of this statistic word frequencies can be predicted with the help of Zipf's law. The derivation is based on equation 2.25 of the previous statistic and is conducted as follows:

$$p_r = \frac{f_r}{f} = \frac{A}{r} \tag{2.27}$$

This will be rewritten to:

$$r_i = A * \frac{f}{f_i} \quad \text{and} \quad r_{i+1} = A * \frac{f}{f_i + 1} \quad \text{where} \quad r_i > r_{i+1} \tag{2.28}$$

In the next step the number of words R_i occurring f_i times are computed.

$$R_i = r_i - r_{i+1} = A * \left(\frac{f}{f_i} - \frac{f}{f_i + 1} \right) = \frac{A * f}{f_i * (f_i + 1)} \tag{2.29}$$

By inserting

$$r_{r=1} = A * \frac{f}{1} = A * f = V \tag{2.30}$$

where $r_{r=1} = V$ is the highest rank, the formula for calculating the number of words R_i occurring f_i times can be rewritten.

$$\frac{R_i}{V} = \frac{1}{f_i * (f_i + 1)} \tag{2.31}$$

Therefore it can be concluded that the fraction of words that appear only once is 50%.

2.3.2.5 Number Of Distinct Words

This statistic refers to *Heap's Law*. It should answer the question of how the size of the vocabulary grows with the size of the corpus. The vocabulary of a text is stated by V with a word size of n and two constants K and β where $10 \leq K \leq 100$ and $0 < \beta < 1$. Both constants are text-depended. (Baeza-Yates & Ribeiro-Neto, 2011; Manning et al., 2008)

$$V = K * n^{\beta} \tag{2.32}$$

The number of distinct words is set by a constant for a language. However, based on the fact that a vocabulary can contain typos, new names or new words it is said that there is no real upper limit. (Baeza-Yates & Ribeiro-Neto, 2011)

2.4 Summary

This chapter contains relevant information for three parts that are essential for the implementation of the prototype in this thesis. Concepts of information retrieval and closely related to that text statistics are both topics that should be transfered by e-learning. E-learning already plays a huge role in the area of teaching which is due to the progress and rapid development of technology on one hand and on the other hand the demand of just-in-time training with purpose to impart knowledge (Uden, 2012). Information retrieval concerns the retrieval of information from a collection of documents. The results of the retrieval process targets the contentment of the user (Baeza-Yates & Ribeiro-Neto, 2011). Basic models in the retrieval process are the boolean model, vector space model and the probabilistic model (Baeza-Yates & Ribeiro-Neto, 2011). Text statistics give information about the behavior of tokens in a text and are used in the process of information retrieval (Manning & Schütze, 1999). An application could be the estimation of the vocabulary size (Baayen, 1996). Furthermore they provide a guide on how to handle certain sources. Statistics like word frequencies in a document as well as in an entire document collection are commonly used in the vector space model for indexing (Baeza-Yates & Ribeiro-Neto, 2011).

3 Related Work

The previous chapters gave an overview of the problems we are facing in e-learning as well as the background this thesis is dealing with. Also the problem this thesis relies on was introduced. The background chapter specified the topics that were covered and implemented in the prototype. Those topics can be divided into three parts which include e-learning, the process of information retrieval and text statistics.
Information retrieval (IR) is a necessary task these days and therefore knowledge of how to build systems for IR is inevitable. When it comes to education in this field, often new approaches can influence the learning process tremendously. This is where e-learning comes into play. Due to the progress of technology and the evolution of the web (Yassine, Khalifa, & Franck, 2017) on the one hand and on the other hand based on the need of distant learning as well as just-in-time learning, e-learning creates an alternative way in education (Rodriguez, 2007). It is also essential in the field of education when it comes to support for situations in which traditional face-to-face teaching is not the appropriate way or simply not possible. Many institutions already provide e-learning as an alternative or additionally (Pamfilie, Onete, Maiorescu, & Pleşea, 2012). The sections shown in this chapter should give an overview of related work regarding tools and software in the field of STEM-education (Bybee, 2010) and especially in the field of information retrieval. Presented literature gives a brief summary of both the same area and also applications in similar areas which served as foundation for ideas in this thesis.

3.1 Related Work for E-learning in Information Retrieval

Apart from web tools like Moodle (section 2) and other content management systems, special software tools for teaching relevant matter in the field of information retrieval were created due to special requirements (Efthimiadis, Fernández-Luna, Huete, & MacFarlane, 2011). Some of them use gamification to some extend. Gamification is an educational approach and method that involves game-elements in non-game tasks in order to boost the learner's motivation and engagement (Phillips, Johnson, & Wyeth, 2013). As the following examples will show, some related research has been performed in this area. Even though some of these systems do not seem to fall in this category at first sight, a great number exploit the fun-factor. Many systems arose from research projects at universities, but are hardly documented for the purpose of self-education. Existing education tools for IR can be divided into two groups, query-based systems and component-based systems (Wilhelm-Stein, Kahl, & Eibl, 2017). Figure 3.1 shows an overview of the tools presented in this section.

3.1.1 Query-Based Systems

Query-Based Systems are systems that focus on query formulation which results in different system behaviors. Basically those systems assess the influence of various search queries. The evaluation of queries using distinct languages represent a special case. Compared to component based systems, users do not get straight control of the information retrieval procedure and IR components are not that important. (Wilhelm-Stein et al., 2017)

IR-Game

In Sormunen et al. (1998) the authors describe the *IR-Game*, a WWW-based tool for playing with principles in information retrieval. Aside from the teaching part it was also used for evaluation of query phrasing especially when it comes to various languages. It is built upon a plug-in based architecture which makes it easy to exchange components in order to alter

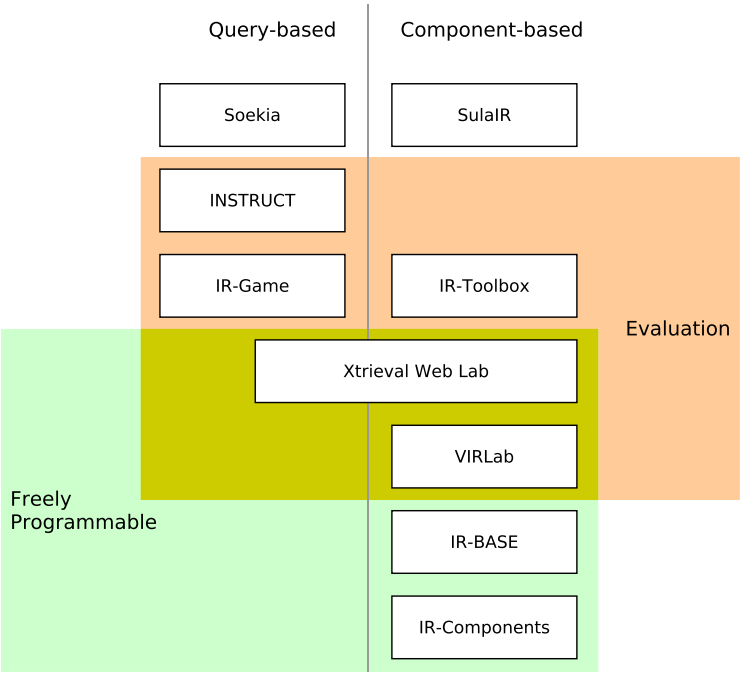

Figure 3.1: An outline of teaching software for the IR area and their capabilities. Image taken from Wilhelm-Stein, Kahl, and Eibl (2017).

its functionality. The tool provides instant feedback on the performance of the query and the system. It also offers automatic generation of graphs like the presentation of precision-recall, as well as a visualization of the result set. Text collections, text retrieval systems, translation dictionaries and morphological programs are the general parts of the IR-Game.

Soekia

This tool was used at lower grade school level and also at college level to help students understanding the mechanics of search engines. Its dimension allows to play with the retrieval process on a very basic level which is expedient only for small data collections. That on the other hand makes it easier to experiment with parameters like the term-frequency in a text. Due to its basic range of functions, it does not include tools for evaluation.

Results are presented in terms of simple HTML-pages (Wilhelm-Stein et al., 2017; Jurjević, Stöcklin, & Hartmann, 2009)

INSTRUCT

In Hendry, Willett, and Wood (1986) the authors describe an interactive tool which should serve as an aid for teaching methods in information retrieval. The main purpose was to use it for students in libraries and information science where knowledge from the information retrieval domain should be gained. This particularly applies to query formulation. *INSTRUCT* contained a text-based and menu-guided user interface. Menu entries comprised a detailed description. Queries can be entered in terms of the boolean model or even by natural language. Considering the time the tool was built it was already packed with all necessary functions like NLP, stop-word-list, stemming algorithm, fuzzy matching, automatic term weighting and automatic relevance feedback.

3.1.2 Component-Based Systems

Component-Based Systems focus on the parts of the IR system itself. The purpose of such systems refers to the idea of the reusability of prior created components. Based on this, some systems allow to create a search engine without the need of knowing to program. This methodology is tendentially used more and more in the development of software since it also improves its maintainability. (Zaitseva, Bule, & Makarov, 2013)

IR-Toolbox

Efthimiadis and Freier (2007) presented the *IR-Toolbox*. This is a web-based learning tool that allows students to build and experiment with their own search engine without any programming knowledge. It aims to teach the IR-process by visualizing it. *IR-Toolbox* uses Apache Lucene[1] as a foundation. Students can play with predefined components by configuring them individually. There is a limitation in the application area

[1] Apache Lucene https://lucene.apache.org/, accessed 2017-12-02

since components can not be placed freely. The learning process takes place in an interactive manner. The *IR-Toolbox* was created for students with a non-technical background, however, since Lucene is an open source library students with technical background can go deeper and explorer its nature.

SulaIR

With *SulaRI* Fernández-Luna, Huete, Rodríguez-Cano, and Rodríguez (2012) created a tool similar to the IR-Toolbox. The learning aid is a desktop application that was built by the use of the Java programming language. It also focuses on the visualization of the IR-process in order to impart the procedure. Even tough the project is delineated as an open source project, the source is not accessible for the public (Wilhelm-Stein et al., 2017).

VIRLab

VIRLab is a web-based tool created by Fang, Wu, Yang, and Zhai (2014) for experimenting with information retrieval models. It works interactively and allows an easy realization of components. Models can be created by the use of a several lines of a language similar to C/C++. The tool makes it possible to change several parameters very easily. *VIRLab* includes test collections that can be taken to evaluate the models. Alternatively one can also create queries. In order to compare different models, charts and metrics can be used. Since *VIRLab* runs on a server, no initial setup is necessary and the focus can be put entirely on the analysis of retrieval models. (Wilhelm-Stein et al., 2017)

IR-BASE

IR-BASE is an object-oriented framework created by Calado, Cardoso-Cachopo, and Oliveira (2007). It aims to be used as an aid in the education process of information retrieval. The framework targets the implementation of new IR components. Architecture and setup are well documented and a guideline leads the student in order to guarantee compatibility among

components. The workflow in *IR-BASE* can be broken up into two parts. First, a fundamental set of components has to be created. Secondly, these elemental items are then used to develop advanced and more complex components. Since this framework does not include functionality to assess validity of new components as well as performance measure, it may lead to undiscovered errors and in further consequence to misbehavior. (Wilhelm-Stein et al., 2017)

IR-Components

López-García and Cacheda (2011) built a Java-based framework that is used to create applications for information retrieval. Basic components for each task are already included. Those have to be linked together in order to form the retrieval procedure and to experiment. It is intended by the authors that students produce their own components to enhance the system performance. However, default implementations can be used, but only as binaries. Contrary to other systems, knowledge of Java is necessary.

3.1.3 Query-Component-Based Systems

Taking the previous mentioned systems as a foundation, a more advanced approach can be observed in Wilhelm-Stein et al. (2017). The project presented in this paper shows a combination of query-based systems and component-based systems. It is a web-based application that focuses rather on creating a retrieval system than on applying programming languages. This should overcome the case when students do not come from a computer science related field and do not know programming. Hence, this way demystifies experimentation. *Xtrieval* allows novices to acquire understanding in the field of information retrieval by playing with a variety of components and real world data in a game mode. This should assure the learners motivation. For the accomplishment of experiments, no set up of a retrieval system in advance is required. This is because the tool already brings along the required environment. The architecture which is client-server based makes the application feel responsive and dynamic. *Xtrieval* contains a web-interface that was written in Javascript and is used to interact with the student. Processing intensive tasks are executed at the

server so the user does not feel any interruption. The back-end runs Java. This tool does not limit the operator to a default workflow.

	QB	CB	FB	GE	PI	WB	PK	FP	E
Soekia	x				x				
INSTRUCT	x						x		x
IR-Game	x		x	x	x	x			x
Xtrieval Web Lab	x	x	x	x	x	x		x	x
IR-Toolbox		x	x		x	x			x
SulaIR		x			x				
VIRLab		x			x	x	x	x	x
IR-Base		x			x			x	
IR-Components		x			x			x	

Table 3.1: Overview of presented learning tools in the field of information retrieval. Abbr.: QB: Query Based, CB: Component Based, FB: Feedback, GE: Game Elements, PI: Platform-Independence, WB: Web Based, PK: Programming Knowledge Mandatory, FP:Freely Programmable, E:Evaluation

3.2 Computer Aided Learning in Related Areas

Through e-learning it is possible to adapt the learning process to an individual. This makes it easier for the learner to acquire knowledge (Mikolajczyk, Moldovan, Chalupczak, & Moldovan, 2017). Apart from education in the area of information retrieval, tools for teaching are also applied in other fields.

NeuroNER

In Dernoncourt, Lee, and Szolovits (2017) the authors present *NeuroNER*, a web-based tool for Named Entity Recognition (NER). The purpose of NER is to tag important entities in a text automatically either by applying rules or by using artificial neural networks (ANNs) which are more efficient (Nadeau & Sekine, 2007). ANNs are especially for verdant users hard to understand. With *NeuroNER* this problem should be eliminated

since it provides a simple tool for NER on the basis of ANNs. Through its graphical user interface it is possible to annotate parts in a text which are then used to train the ANN. The tool allows users to play with annotations while either existing or newly created collections can be applied. Created annotations can always be altered so that one can get and improve insights of the procedure. The training task can be supervised in real-time. Resulting outputs contain text as well as visualizations like classification reports and learning curves of the learning process.

Parson's Programming Puzzle

Parsons and Haden (2006) present an interactive tool for introducing the concept of programming. *Parson's Programming Puzzle* is web-based and includes, as the word puzzle may already indicate, game elements in the form of a puzzle. This style should increase and keep the student's motivation and participation at a good level. A puzzle states a problem. Every piece of the puzzle contains a description in either written format or as activity diagram that show the process. The tool uses a drag-and-drop model in order to place each part of the puzzle at the right location. Every action can be evaluated. Students have to improve their solution until they reach 100 percent correctness. This assures that code is not just implemented to get the right output but also forces an appropriate code design. Students are allowed to make errors, because making errors is part of the learning process, but they should also get the possibility to choose between incorrect and correct options.

Database Normalization Tool

In Stefanidis and Koloniari (2016) the authors describe a software tool developed in C# that should be used to teach and learn how database normalization works, but also acts as an ordinary tool that can be smoothly applied to small project projects in order to automate the normalization process. The tool requires a database table as input and a set of BCNF functions. As the database learning tool is used for teaching, it has the ability to provide help during the presentation of a normalization process. When it comes to self-teaching, users can easily experiment with each

step of the normalization procedure through the graphical user-interface. Each step can be affected by changing functional dependencies and related tables. Predefined guides assist the student during the process.

Wolfram Alpha

Wolfram Alpha[2] is "A Computational Knowledge Search Engine" as described in Cassel (2016) developed by Wolfram Research[3]. It is based on Mathematica[4] and was initialized when the Web 2.0 started to burgeon. The author of the paper presents a search engine that distinguishes from what we are used to, for comparison Google. Based on a given query Google would respond with ranked documents regarding to some extend of similarity. Wolfram Alpha instead answers the query by linking facts for computation and responds with adjusted and built-in results. The results contain textual output like facts to a question and visualizations of those facts. Also explanations can be part of the result. Wolfram Alpha offers an easy-to-use web-interface. Users can insert any query. This can be for example a word, fact, mathematical equation or phrase. Since it is also a project that considers artificial intelligence, the search engine can deal with queries entered in natural language. The database of Wolfram Alpha covers mathematics, science and technology, society and culture and the everyday life. Based on its flexibility and wide range of topics this tool suits the requirements of an application in the field of teaching. It allows to play with various inputs and immediate outputs present the user its impacts. Additionally the output can be modified interactively.

Wolfram Demonstrations

Another project created by Wolfram Research is the *Mathematica Player* (Normand, Lesmes, Corradini, & Peleg, 2010). This tool is for free and can be used to play with a huge set of demonstrations. The user can choose from more that 11000[5] demonstrations in many different fields of science,

[2]Wolfram Alpha https://www.wolframalpha.com/, accessed 2018-14-03
[3]Wolfram Research https://www.wolfram.com/, accessed 2018-14-03
[4]Mathematica https://www.wolfram.com/mathematica/, accessed 2018-14-03
[5]Wolfram Demonstrations http://demonstrations.wolfram.com, accessed 2018-14-03

engineering and many others. In Normand et al. (2010) the author describes the application based on an example in food engineering. Every demonstration includes a short description of its scientific background and how parameters impact the demonstration's behavior. Some of them can be even utilized for real computations. The demonstrations were created by professional and amateurs and are aesthetically pleasing. Mathematica Player is a tool that should be used as an aid in education and research.

3.3 Summary

In this chapter related work regarding tools in the field of e-learning were covered. As the chapter shows, not only information retrieval is in the need of such helpers, but also other areas like named entity recognition or database normalization make use of it. However, some of them share the same functionality or methodology in order to provide reasonable teaching support. This includes a feedback mechanism at each step in the process or the freedom of implementing own components. Other tools already bring along prefabricated components so students can start immediately building the target system. Usually such tools do not require an initial set up in order to put the focus entirely on the experimenting part. An important fact is that some of the presented learning tools do not expect previous knowledge in programming which is very positive when an usage for non-computer science students is planned.
Teaching aids are a major topic, especially in the STEM domain and research yields new applications to provide help in the education process. However, the challenging parts are to keep the student motivated during the learning process and of course to transfer the knowledge in a suitable way. Some creators try to tackle this problem by using game elements in order to enhance the students motivation. Others encourage students to play with demonstrations. This especially applies to Wolfram Demonstrations whose development nearly entirely aims in this direction. Nevertheless, nearly all of the presented tools try to put the focus also on a graphical interface that facilitates the utilization. In conclusion, it can be pointed out that most developers of learning tool emphasized platform-independence

and therefore used Java[6] or web-based technology.

[6]Java https://www.java.com/de/, accessed 2018-02-02

4 Design and Development

Due to abundant research of related work, several findings turned out to be important for a tool in education, while none of the reviewed tools seem to cover every feature in order to bring full satisfaction. However, there is a high demand for applications to provide support in teaching and learning situations especially in the STEM area as pointed out in Chapter 1.
This thesis aims to create a platform-independent tool with two key goals. First, learning the essential concepts of information retrieval by building a simple search engine in an active and engaged way. Secondly, investigating and experimenting with text statistics that are relevant in order to understand the behavior of natural text. As seen in Chapter 3, tools for assistance were already built and the feedback for those tools confirmed the creators to be on the right way. Apart from the knowledge transfer through the application, the focus also lies on the user-tool-interaction and therefore the user-interface was a major part when the tool was developed. The importance of that fact is emphasized by Faghih, Azadehfar, and Katebi (2014) who argue that success of an application in the education domain is defined by the style of its user-interface.
The following sections of this chapter will cover the fundamental techniques and ideas used in this thesis to develop a tool for e-learning in the field of information retrieval. Based on related work and regarding the methodology applied in the course, this tool should be developed for (Chapter 1), requirements will be elucidated. This will then lead to the concept and architecture. In the last two sections, used tools and libraries will be discussed and a conclusion will be presented.

4.1 Basic Requirements and Core Concept

Based on a course that covers the fundamentals of information retrieval and provides notes and additional content using a Moodle CMS (Chapter 2) as teaching aid, the primary objective was to develop a tool that integrates interactivity in order to implement the concept of MAL (Pirker et al., 2014). This covers the assembly of a simple search engine and the experimentation with basic text statistics.
The concept for the prototype is spread over two parts - the search engine and the text statistics - and covers at first basic functions which will than be in future versions adapted with advanced functionality. As findings from chapter 3 show, there are some features in teaching tools which go down well with students. These include for a start modularity. A search engine consists of multiple parts. For each of them it should be possible to be created independently. This is achieved by separating steps into modules. Since the modules can be created separately, there has to be a default implementation in order to run the system. Predefined parts bring benefits to one another. It allows the students to use the tool for pure exploration of the retrieval process without the need to write a single line of code. Additionally, the default implementations can be used to evaluate created modules. By this means, the user can choose if the own module or the provided module should be used and can control the data flow between the parts. The second part of the prototype, text statistics, does not depend on the search engine. It provides various statistics based on selected documents. Also the statistics do not rely on each other. Only one statistic can be applied to text at a time. This part of the tool is only for experimenting with documents in order to see the behavior of natural text. Until now, there is no intention to make it programmable, but some modules like the parser or the crawler can influence the result if newly added documents or document collections are used. Visualizations are a major feature in education. Additionally to results in a text format they complement the output of text statistics. Compared to solely text based systems they provide eye candy and so increase the attention of the individual. Hence, much more important is the fact that they support the understanding of results.
As a graphical user interface often eases the handling, there was already at the beginning a strong preference regarding this way of interaction with

the system.

The application should be able to handle both data from online resources as well as local document collections. Either sources can be used by both parts. Online resources, which are in case of a search engine simply web-pages, need to be fetched in order to be used by the tool. The downloaded data is then added to the list of previously available document collections. For the use of web-pages as document collections the prototype provides an additional module, a web-crawler, which is also a part students can implement themselves. There is no difference when accessing the data, because it is the repository except if the local file is going to be used the first time. This leads to the file cache - a necessary helper since document collections can get very large. It make sense to cache data in the memory to increase the performance and to store data binary in the file-system in order to speed up the application start.
Feedback is a vital feature in such kind of applications. This turned out to be one of the key factors as seen in related work. The prototype in this thesis should provide feedback in two ways, depending on the type of action. On the one hand, if the user wants to start a statistic or run the search engine without having selected data, a pop-up is shown with an error message. On the other hand, if the user runs the search engine, one will get immediately output of the current progress in form of text. It may also contain error messages.
Platform-independence for education tools is also a very important aspect. This will be answered in more detail in section4.2. In order to use the tool instantly, offline data is necessary. For this propose it is important to have a provided predefined document collection. Due to the size and simplicity the Cranfield Collection (section4.2) was chosen.
Finally, the tool should act as a teaching aid and in further consequence it should be used for automatic grading. For both purposes, the application will be executed with a *config.xml* - a configuration file that includes preselected documents and settings which provide initializations for the tool. By this means the behavior can be easily changed. Figure 4.1 shows the conceptional architecture. It describes the basic principle of the system. This starts with four inputs. Two of them represent the data sources that are used in the tool, one supplies initial configurations and one depicts the user input. The main part in the middle shows three boxes representing the logic. This includes steps for preprocessing, the retrieval model and

statistics. On the right side, one can see three boxes that state entities for output of the retrieval process as well as for statistics.

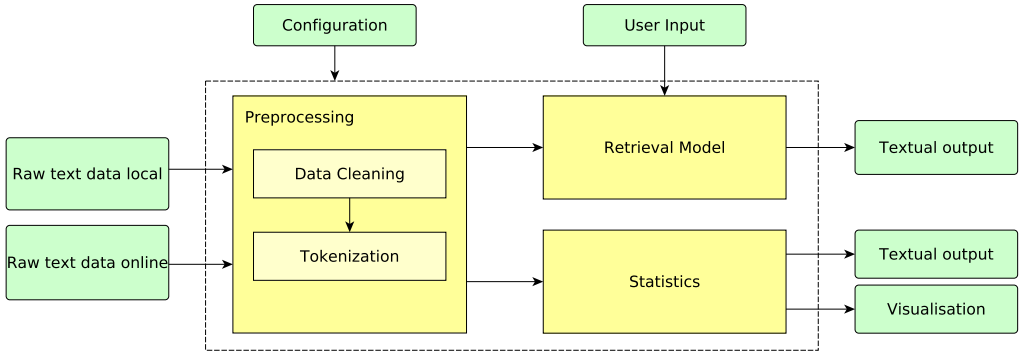

Figure 4.1: Conceptional architecture of the prototype - ISRApp.

4.2 Technologies and Libraries

This section addresses technologies and libraries used in the prototype to successfully accomplish the requirements presented in section 4.1 in order to create a reasonable architecture. Platform-independence was a inevitable demand and led to two main options, platform-independent desktop applications or web-based development. For the first prototype, Java 4.2.1 was chosen since it already provides a lot of functionality and performs well compared to web-based applications. The development was performed using the Eclipse[1] IDE because of its simple but powerful toolset and its easy extensibility using plug-ins. This leads to JFormDesigner and WindowBuilder 4.2.2 which are extensions for easy GUI development. The number of freely available libraries for creating visualizations is manageable, but a great alternative was found in JFreeChart 4.2.4. Many open source cache libraries are available. Due to its advanced development and very detailed documentation, Ehcache 4.2.3 was chosen.

[1]Eclipse https://www.eclipse.org/, accessed 2018-04-09

4.2.1 Java

Java[2] technology was used as programming platform. This was due to a demand on independence of operating-systems, but still allows to build applications with a comparable good performance.

4.2.2 JFormDesigner and WindowBuilder

JFormDesigner[3] and WindowBuilder[4] are both extensions for major IDE's to ease the development of graphical user-interfaces (GUI) in Java. GUIs can be created using drag-and-drop in order to place elements or by modifying the source code.

4.2.3 Ehcache

Ehcache[5] is an open source cache for Java available under the Apache open source license. It can be used for distributed caching and supports caches in memory a well as disk drives. Due to very few dependencies, existing applications can be easily extended.

4.2.4 JFreeChart

JFreeChart[6] is an open source Java library for creating charts. It covers a broad number of chart types and comes with a well-documented API. JFreeChart can be easily extended and runs on client- and server-side software. The library includes functions to adapt the visualization to the data like zooming or printing and saving to the file system.

[2]Java https://www.java.com/, accessed 2018-03-14
[3]JFormDesigner https://www.formdev.com/, accessed 2018-03-14
[4]WindowBuilder https://www.eclipse.org/windowbuilder/, accessed 2018-03-14
[5]Ehcache http://www.ehcache.org/, accessed 2018-03-14
[6]JFreeChart http://www.jfree.org/jfreechart/, accessed 2018-03-14

4.2.5 Cranfield Collection

The Cranfield Collection[7] is a standard test collection containing 1400 abstracts of aerodynamics journal articles (Manning et al., 2008). It comes with the prototype and is intended to be used as offline data source.

4.2.6 Development Environment

The development of the prototype was performed on both Windows and Linux systems. Most of the time Java from Oracle[8] was used for development, but for compatibility reasons the application was also tested with the OpenJava[9] environment. The following list shows an enumeration with all used tool and their exact version number in order to continue the development without compatibility issues.

- Ubuntu Linux 16.04
- Windows 7 x64
- Java 1.8, OpenJava 1.8
- JFormDesigner 5.2
- WindowBuilder 1.8.2
- JFreeChart 1.0.19
- Ehcache 2.10.1

4.3 Architecture

Based on the concept and the design decisions regarding used technologies in the previous sections, this section will cover the chosen architecture for the prototype. This section explains the structure of the application named in the following *ISRApp*. Figure 4.2 shows an overview which consists of four groups. The purpose of the segmentation is to immediately see parts that belong logically together. Those groups are:

[7]Cranfield Collection http://ir.dcs.gla.ac.uk/resources/test_collections/cran/, accessed 2018-03-14

[8]Oracle https://www.oracle.com/, accessed 2018-04-25

[9]OpenJava http://openjdk.java.net/, accessed 2018-04-25

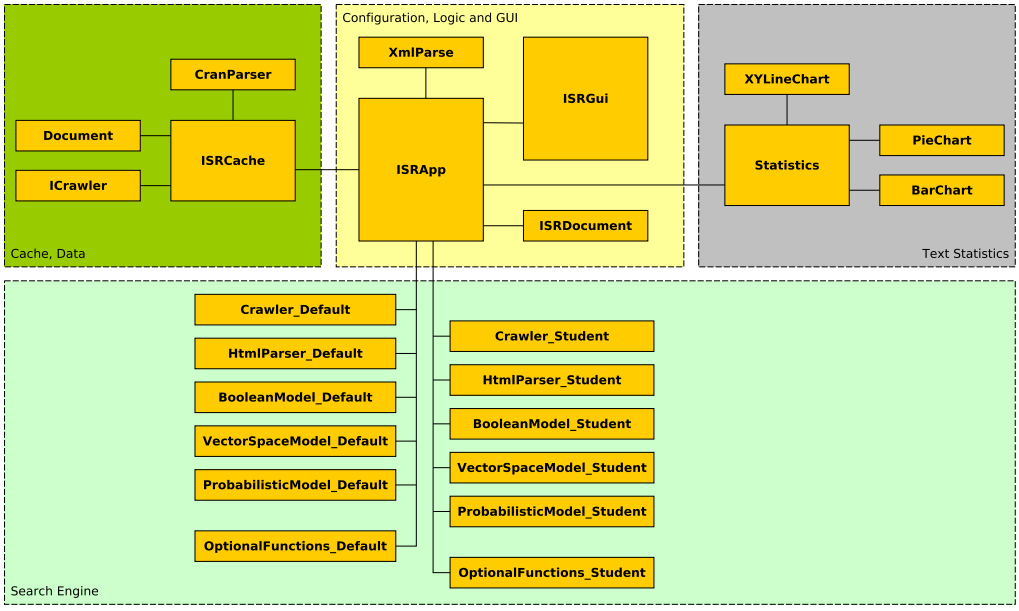

Figure 4.2: Architecture overview of the prototype - ISRApp.

- Configuration, Logic, GUI
- Cache, Data
- Search Engine
- Text Statistics

In the following sections the purpose of those groups is explained as well as their subparts. Since the illustrated modules represent classes of a programming scheme, the following explanations will rather refer to classes.

4.3.1 Configuration, Logic, GUI

In this group the primary components of the ISR-App are covered. It includes the main class ISR-App, the class for the user interaction - ISRGui and XmlParse - a class that reads the configurations.

4.3.1.1 ISR-App

This is the main entry point of the prototype. It is the largest class and handles the main process. This includes configurations which are set up at startup, the initialization of the graphical user interface, the start of the cache module which holds or loads the data, the execution of the crawler, the invocation of the blocks and the statistics part.

4.3.1.2 XmlParse

XmlParse reads configuration entries at the beginning from the file system using a configuration file. This includes document collections as well as settings for the application. The data is then accessible for the ISRApp class.

4.3.1.3 ISRGui

The ISRGui-class provides the graphical user interface for the prototype. This is divided into input and output section. The input part can be used to set up the search engine or to select a statistic. Search engine and Statistics are two tabs in the input area. Beside that an extra tab is located next to them. This points to optional functions, which are implemented but not yet connected to the system of the tool and therefore deactivated. The output section is a simple text-field that gives the user information about the process or simple error messages. On top of the main frame there are sub-windows for the crawler and the data selection. Each of them can be called with a separate button. Additionally, the class supplies the user with error pop-ups.

4.3.1.4 ISRDocument

ISRDocument is the final representation of a text file or HTML file. It is an internal data structure that eases the file handling while it unifies different types of files. This kind of document already contains attributes for easy access of additional information like, title or author.

4.3.2 Text-Statistics

This group of classes is used to provide the user statistic functions which can be applied on textual data sets.

4.3.2.1 Statistics

In this class text statistics are calculated. The data is set in the ISRApp when this part is activated. Since visualizations are an important supplement to textual results, each statistic supplies the user with plot functions using various kinds of charts. Those will be listed hereafter.

4.3.2.2 BarChart

The BarChart is a class using the JFreeChart library in order to show the user bar-charts for given data.

4.3.2.3 PieChart

The PieChart is a class using the JFreeChart library in order to show the user pie-charts for given data.

4.3.2.4 XYLineChart

The XYLineChart is a class using the JFreeChart library in order to show the user xy-line-charts for given data.

4.3.3 Cache, Data

The group in this section concerns file managing and providing data. Therefore, a cache system is implemented to support for big data collections. It consists mainly of four parts. The ISRCache, which is the main class, CranParser a parser for the Cranfield-Collection, an interface for the crawler and the Document class - a simple file mapper.

4.3.3.1 ISRCache

This class is used to cache data. It contains the cache logic and holds the data that is used in the search engine and for statistics. Especially when large web-sites or collections are used, it makes sense to store it not just in the memory but also in the file system in a binary structure. Thereby the start-up or restart of the application can be increased a lot.

4.3.3.2 Document

The Document class is a simple container that maps the content of a text-file (e.g. Cranfield-text) or a HTML-file. This is also the part that is used when it comes to serialization of data in oder to create an image in the file system.

4.3.3.3 CranParser

CranParser is a simple parser for documents of the Cranfield collection. Its result is a list of Documents.

4.3.3.4 ICrawler

This is the interface for the crawler which finds its implementation in the Search Engine group which will be covered in the following part.

4.3.4 Search Engine

Search Engine is the group that contains the modules to be implemented. Additionally, each module comes with a version that is already implemented.

4.3.4.1 HTMLParser

This class is used to parse simple html-pages.

4.3.4.2 BooleanModel

The BooleanModel class provides the index and the search functionality for the boolean model.

4.3.4.3 VectorSpaceModel

The VectorSpaceModel class provides the index and the search functionality for the vector space model.

4.3.4.4 ProbabilisticModel

The ProbabilisticModel class represents the search functionality using the probabilistic approach.

4.3.4.5 OptionalFunctions

OptionalFunctions is a placeholder for functions that are already implemented, but not yet integrated in the system of the tool in order to use it for experimentations. This includes the following list of functions which were defined in chapter 2:

- Inverted Index
- Levenshtein Distance

- Query Expansion
- Fuzzy Logic
- HITS
- Page Rank

4.4 Deployment and Usage

In this section the prototype will be explained from a user's point of view. At first one can find instructions for the installation, then the usage of the ISR-app will be explained. It consists of two main parts. The search engine and text statistics. The search engine provides the possibility to crawl pages or use local stored documents on which retrieval models can be applied. This covers the boolean model, the vector-space model and the probabilistic model. Html files have to be parsed and therefore a parser is included. Each of the modules will be shown as well as their usage. Statistics do not provide as much functionality as the first part, but may still need some introduction.

4.4.1 Installation

This subsection explains how to install the ISR-app in two ways. The application can be run within an IDE or executed as a standalone experimentation tool. When the first option is chosen, the provided folder containing the eclipse-project can simply be imported. Figure 4.3 shows the project structure. Option two assumes that the application is only used as standalone program for playing with text statistics due to the state of development. For this case, a jar-file is included in the project folder. It contains all the binaries and libraries so the application can be launched easily without an IDE. In figure 4.4 the folder structure is shown. However, to run the ISR-app following files are important and must not be removed or moved to other locations:

- Binary classes or the .jar-file
- "config.xml" configuration-file which contains all the settings
- Configuration-file for the cache: ehcache.xml

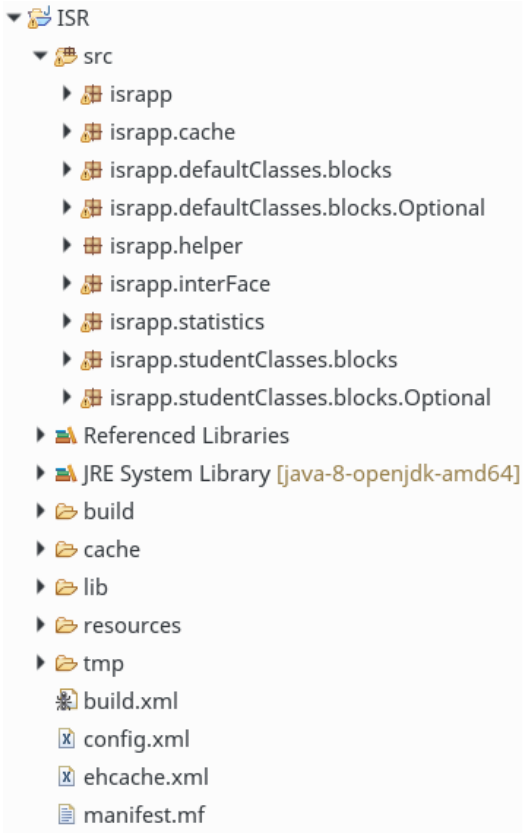

Figure 4.3: Folder structure of the prototype.

- Resource folder
- Cache folder (will be created if not available)

After the start of tool, the screen in figure 4.5 is shown which consists of two parts - a control part which lets the user manage the input and an output part which is a simple text area. At this time the configuration file has already been loaded. The user can provide its own configuration at the application's start. If there is no configuration file specified, a default configuration will be taken instead.

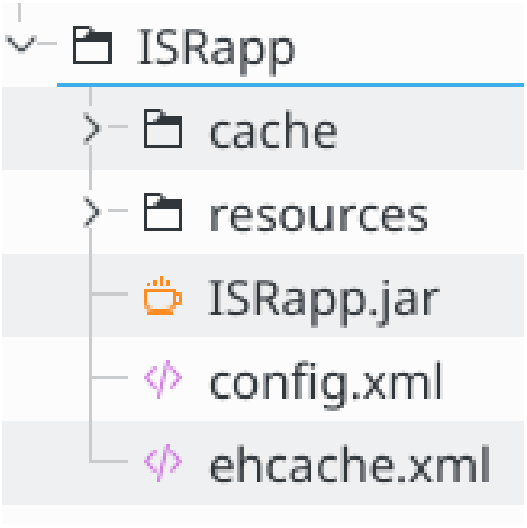

Figure 4.4: Folder structure of the standalone version.

4.4.2 Search Engine

For demonstration of the search engine, a very small test collection was used to illustrate the entire set of functionalities. In order to use the search engine functionality, the "Search Engine Tab" has to be selected. This results into figure 4.6. This tab is bipartite and shows the user at the upper half a brief description of how to use the tool. The lower half gives the user the possibility to select the desired components, an input text-field for the query and the "Run-Button" to start the retrieval process. The next step is to select one or more collections using the "Select Collection-Button" which can be found in the top left corner in figure 4.6. Clicking the button opens a new window which shows all available collections and their number of documents when marking them as well as a total number of documents since more than one collection can be used. This is shown in figure 4.7. After selecting documents, the window can be closed. At this point the user needs to choose the mode for the run. There are three modes that can be selected. A default mode which is used for default implementations indicated by the letter "D", a student mode to execute modules implemented by the user and presented using letter "S" and a grade mode which is an additional feature to use the tool in further consequence for grading. The grade mode is indicated by the letter "G".

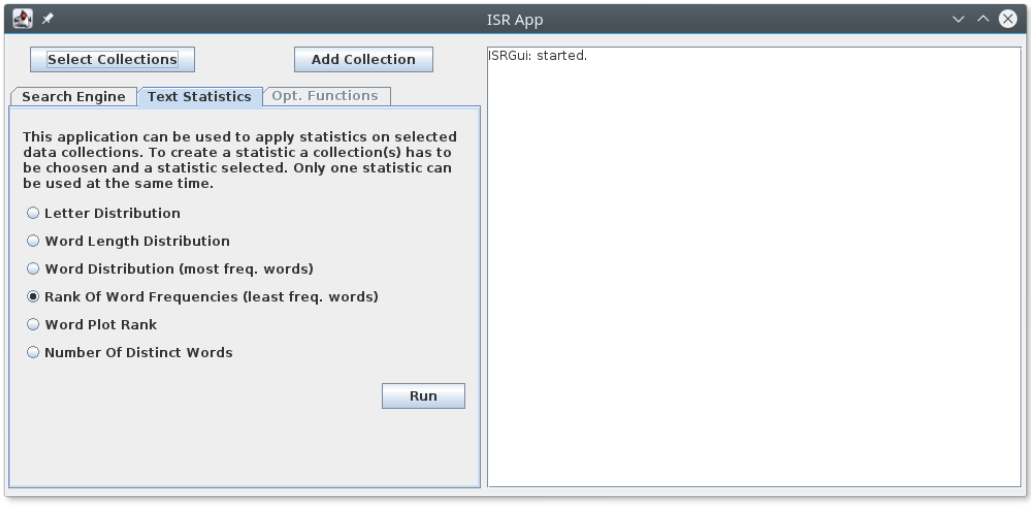

Figure 4.5: View of the prototype after start. The entry point of the tool shows the statistics tab.

If the grade mode is selected, no other setting can be made. Apart from the grade mode, the user can mix the modes and select different modes for different modules. This allows to experiment with the tool using own implementations even though some modules are not implemented by the user. After the document selection the parser has to be ticked "Html Parser" in order to proceed. The last setup step is to choose the desired retrieval model. Only one model can be chosen at a time. Now the query can be entered and a click on the "Run-Button" starts the retrieval process. Basically the selection of the document collection and the query input can be altered at any time. The only thing that is important is to provide data in order to start a run. Otherwise the tool will come up with an error message as a reminder as figure 4.8 shows.

4.4.2.1 Crawler

The crawler allows to extend the document collection by saving web-pages to the file system. In order to get to the window of the crawler which is shown in figure 4.9 one needs to click on the "Add Collection-Button". As the figure presents, there are a few options to be set. These include the crawl-url, a maximum number of crawl-width, a maximum number of

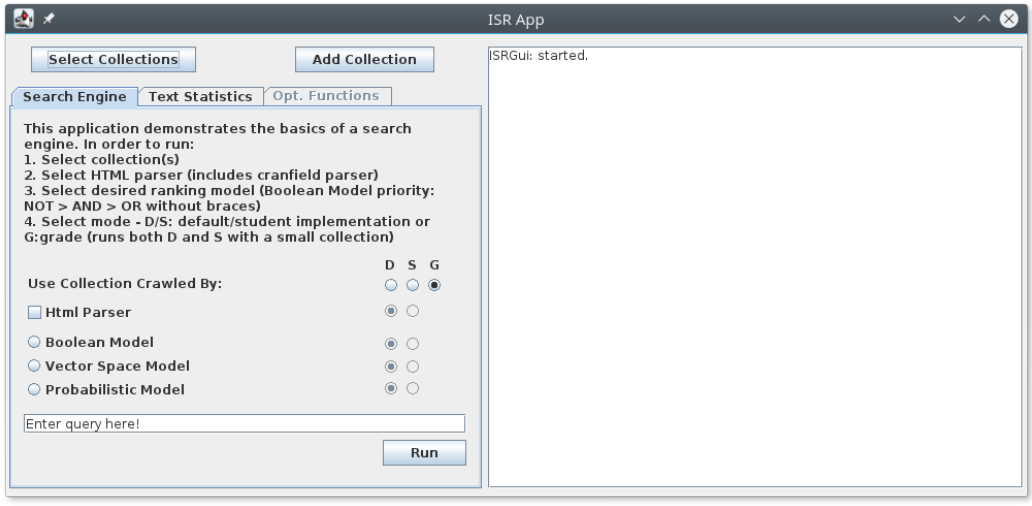

Figure 4.6: View of the prototype search engine.

crawl-depth and a number for server requests to prevent the crawler from failing when it comes to timeouts. Since the crawler is a module which can also be implemented by the user, the option "crawler mode" gives the possibility to switch between default- and student implementation.

4.4.2.2 HTML Parser

The HTML Parser is a simple parser for crawled web-pages or local html-files. Due to the design of ISRApp the parser has to be ticked when using the search engine even if documents are not in html format. Error message when not selected.

4.4.2.3 Boolean Model

The Boolean Model is a very simple implementation of the boolean retrieval model. It makes use of a term-document-matrix (section 2) for indexing and for the search process. Small document collections (≤ 50 words) allow to print the index so students can take a look at the data structure. In figure 4.10 the output of the boolean model can be seen when a small

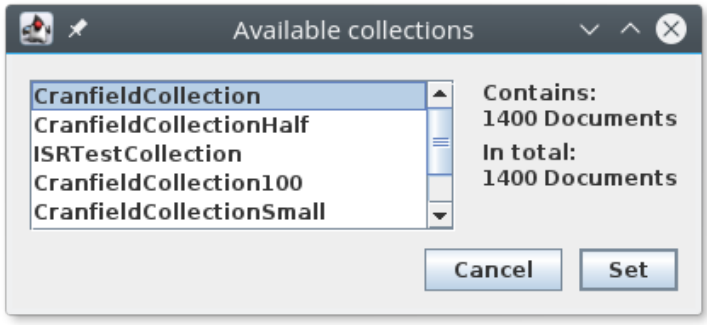

Figure 4.7: Overview of available collections to choose.

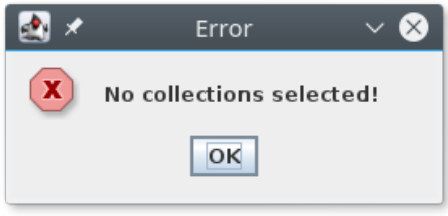

Figure 4.8: View of the prototype after start. The entry point of the tool shows the statistics tab.

document collection is used. This includes the term-document matrix as well as retrieved documents for the query "user OR model".

4.4.2.4 Vector Space Model

The vector space model allows the user to retrieve documents using a search mechanism for result ranking. Like the boolean model it uses a term-document matrix for indexing but with weighted entries and provides its presentation for small (≤ 50 words) document collections in order to gain insights when experimenting with this model. This is shown in figure 4.11. In output area of the tool, the user is given the formated buildup of the data structure using weighted entries and a ranked result list for the query "user model".

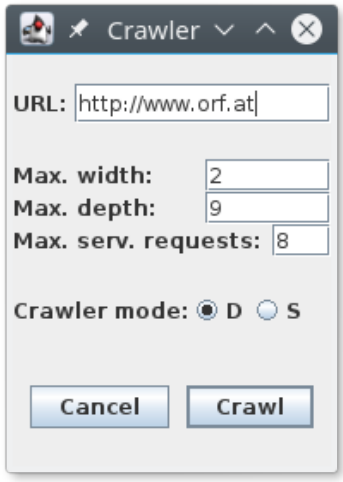

Figure 4.9: Crawler view of the prototype. Parameters like crawl-depth, crawl-width and max. server requests can be provided.

4.4.2.5 Probabilistic Model

Like the vector space model, the probabilistic model also provides a ranked result set. However, the way it works requires a different approach. This includes an initial ranking which is based on the inverse document frequency and uses in further consequence several iteration steps. The number of documents is used to adjust that. Figure 4.12 shows the procedure for the query "user model". It sets initial probability values for the keywords in terms of relevance or irrelevance and provides an initial document ranking using the inverse document frequency. Based on that, another iteration takes place in order to re-rank the documents regarding their score of relevance.

4.4.3 Text Statistics

This section covers text statistics, the second big part of the ISRApp. Text statistics are an important factor in the retrieval process when it comes to understanding and improving. In order to get to the statistic functions, the tab "Text Statistics" has to be selected. It is usually the first view after the application has started and is shown in figure 4.5. In the following, these

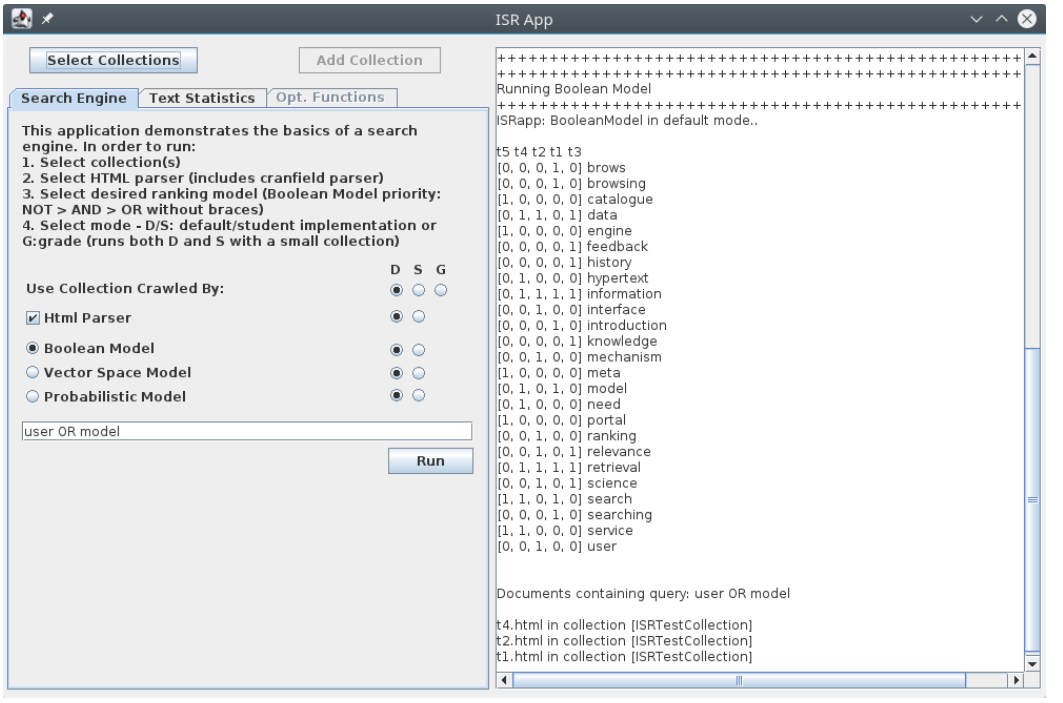

Figure 4.10: Output of the boolean model when used with a small document collection.

statistics are listed and explained which students will be able to experiment with using the tool. Each statistic can be used independent of the others. Only one statistic can be used at a time. The tool will be delivered with a predefined document collection, the Cranfield collection. This serves as the basic data source, but text statistics can also be applied on other documents. The output of a statistic function is text based, but is additionally supported by one or more visualizations. For the demonstrations of the functionality, the documents of the entire Cranfield collection was used. Additionally, each statistic function shows basic information like the selected collections, the total number of words, unique words and the number of documents.

4.4.3.1 Letter Distribution

This statistic shows the distribution of letters in a text. In the output field, one can find the textual output which lists the letters using their rank which is based on the frequency. Furthermore the output gives a percentage of the

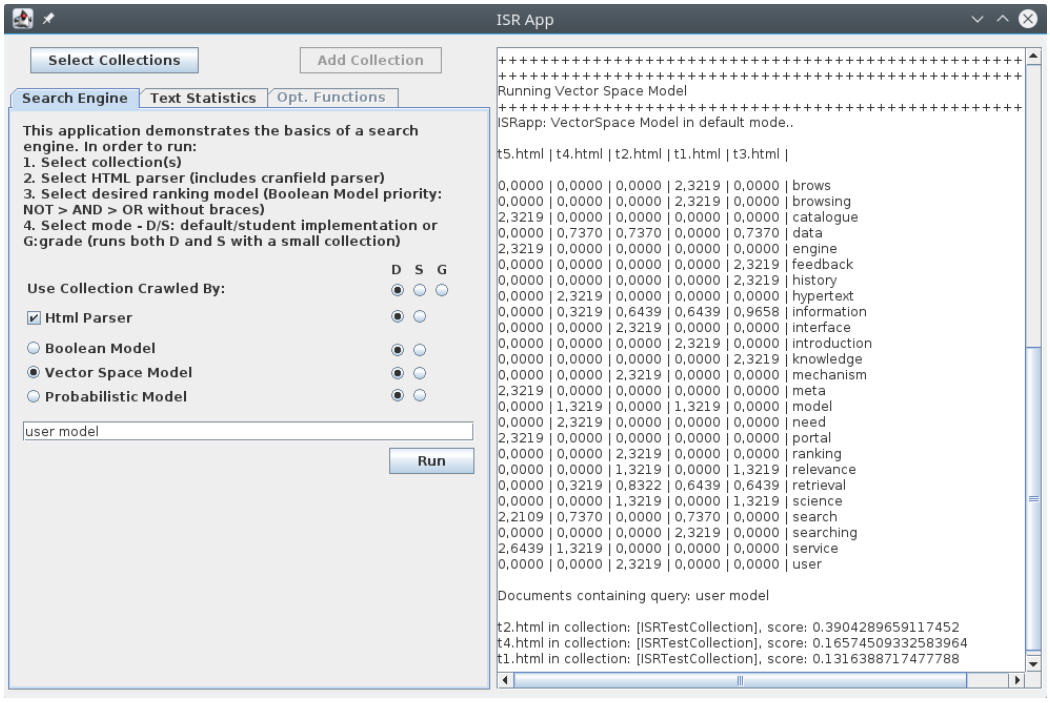

Figure 4.11: Output of the vector space model when used with a small document collection.

letter distribution over the entire collection. This can be seen in figure 4.13. In figure 4.14 one can find the distribution visualized. Figure 4.15 presents the distribution in percent.

4.4.3.2 Word Length Distribution

Using this statistic one can gather information about the distribution of word lengths in a text corpus. The output shows the distribution ordered by the frequency and states also the coverage in percent. Additionally, one can find the average word length. This can be seen in figure 4.16. In figure 4.17 the output is visualized. Figure 4.18 shows the distribution in percent.

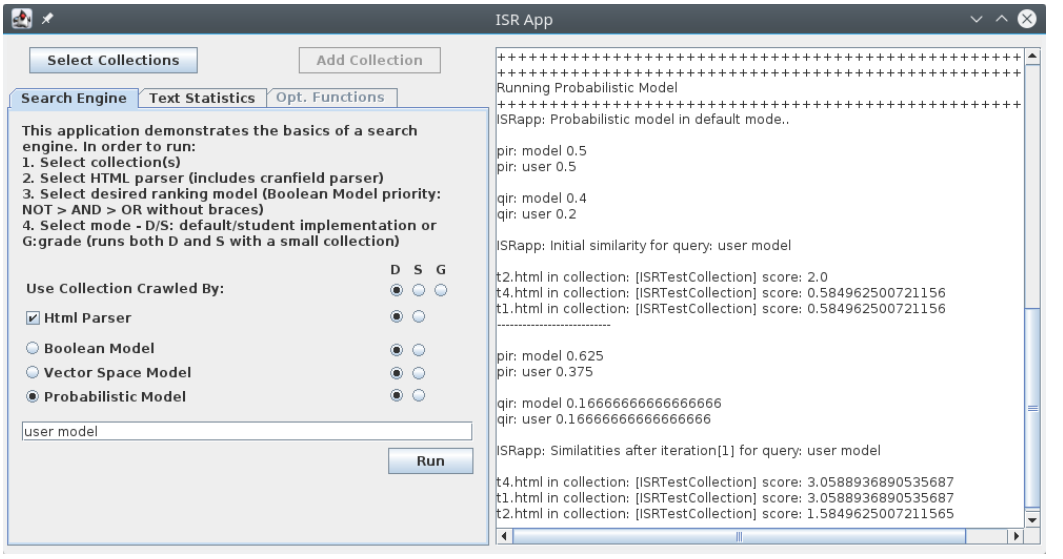

Figure 4.12: Output of the probabilistic model when used with a small document collection.

4.4.3.3 Word Distribution (most frequent words)

Word Distribution gives information of the word coverage in a document collection. In the output, words are ranked based on their frequency. It also contains the percentage and an additional column that shows the relation of the rank and the percentage for natural text and is a constant. All outputs from this statistics are limited to 50 words in order to keep the focus to a reasonable size. This can be seen in figure 4.19. Figure 4.20 visualizes this output and in figure 4.21 one can see the visualization in percent. An additional figure 4.22 shows the constant that is obtained by the multiplication of rank of a word and the percentage of its coverage.

4.4.3.4 Rank Of Word Frequencies (least frequent words)

The statistic “Rank Of Word Frequencies (least frequent words)” shows the user the frequency of word frequencies. Due to reasons of visibility and reasonableness, the number of word frequencies was limited to 20 words. In the output, one can find a ranked list of occurrences as well as the fraction of the rank of the i-th word divided by the number of words in the

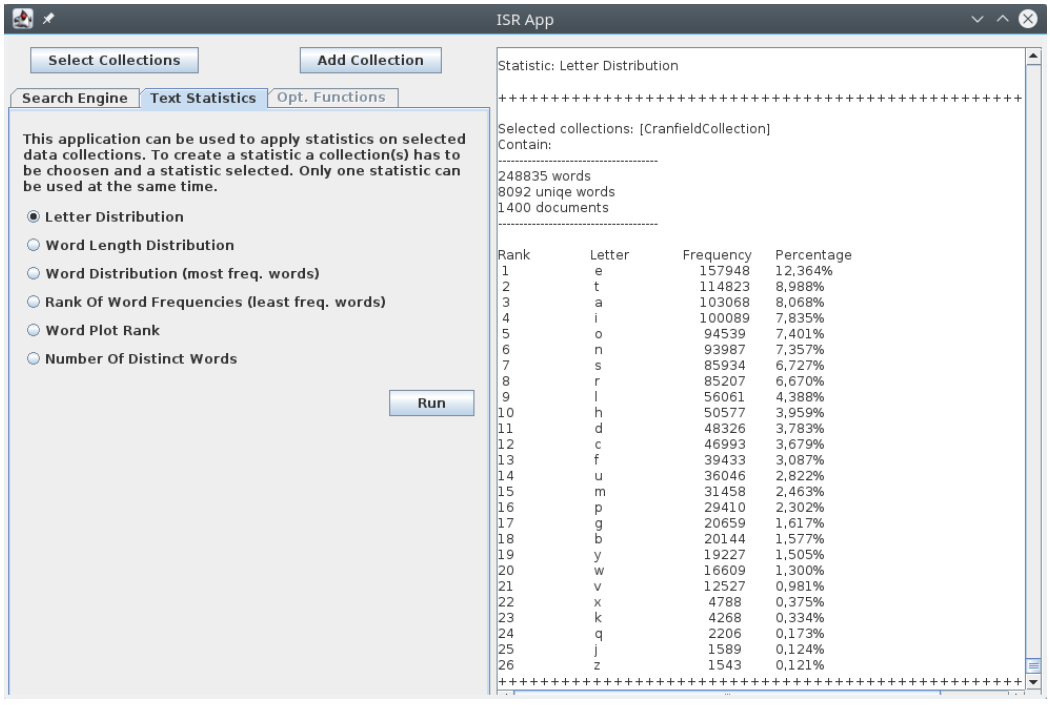

Figure 4.13: Textual output of the statistic "letter distribution" using the Cranfield document collection.

vocabulary. The textual output can be found in figure 4.23. In figure 4.24 the corresponding visualization is shown.

4.4.3.5 Word Plot Rank

By the use of the statistic "Word Plot Rank" one can see the relation of the absolute frequency of words and the word rank in a log-log space. Figure 4.25 presents this behavior.

4.4.3.6 Number Of Distinct Words

The statistic function "Number Of Distinct Words" shows the user the relation between the number of words in the vocabulary and the total number of words in the document collection. Figure 4.26 shows the textual output

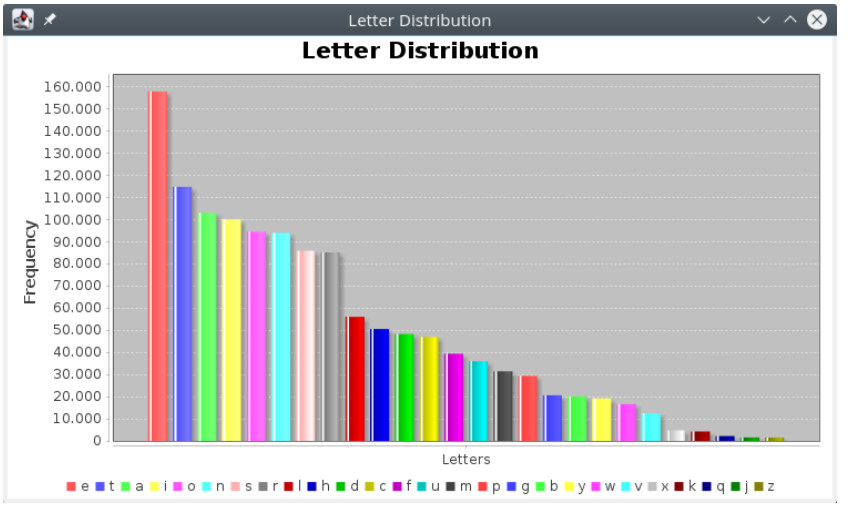

Figure 4.14: Visualization of the statistic "letter distribution" using the Cranfield document collection.

which is limited to the first 30 documents. It can be seen in figure 4.27 that at some point the number of unique words start to convergence.

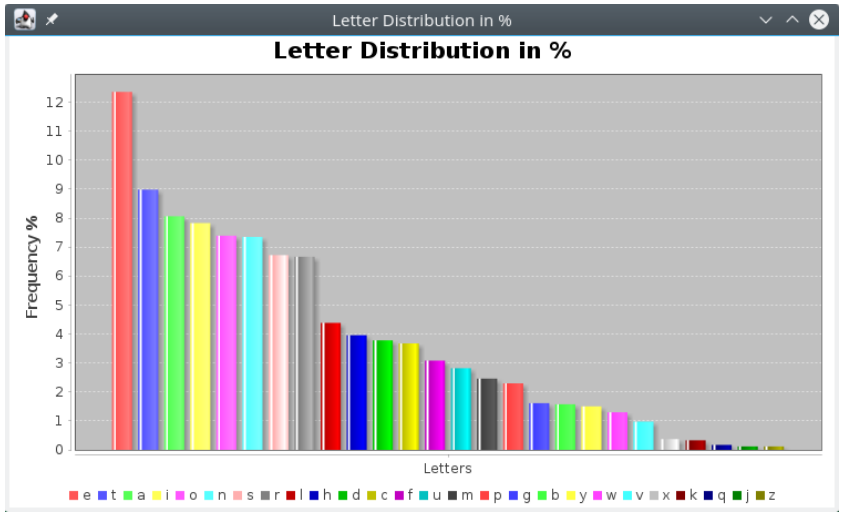

Figure 4.15: Visualization of the statistic "letter distribution" using the Cranfield document collection in percent.

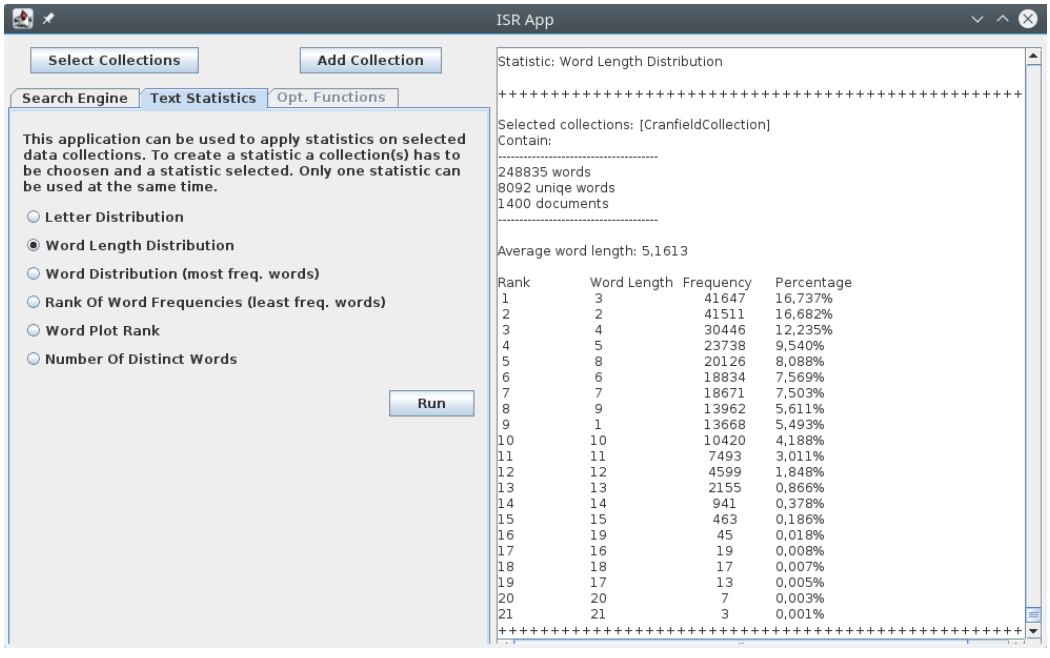

Figure 4.16: Textual output of the statistic "Word Length Distribution" using the Cranfield document collection.

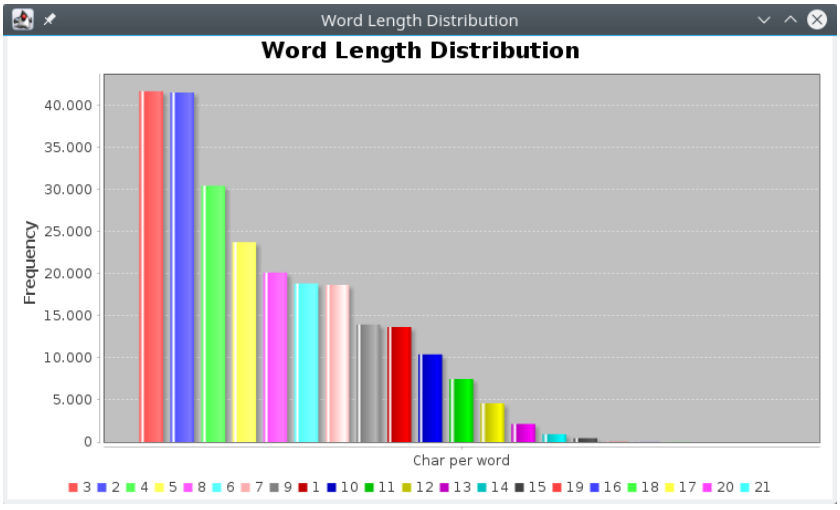

Figure 4.17: Visualization of the statistic "Word Length Distribution" using the Cranfield document collection.

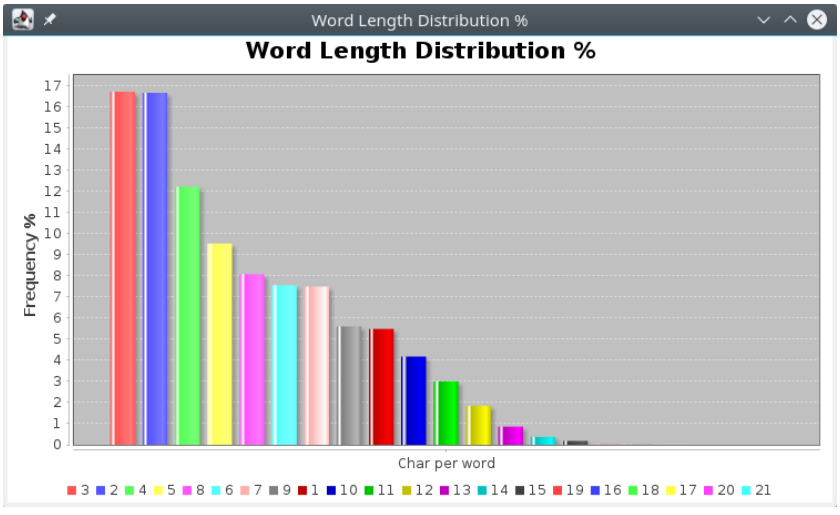

Figure 4.18: Visualization of the statistic "Word Length Distribution" using the Cranfield document collection in percent.

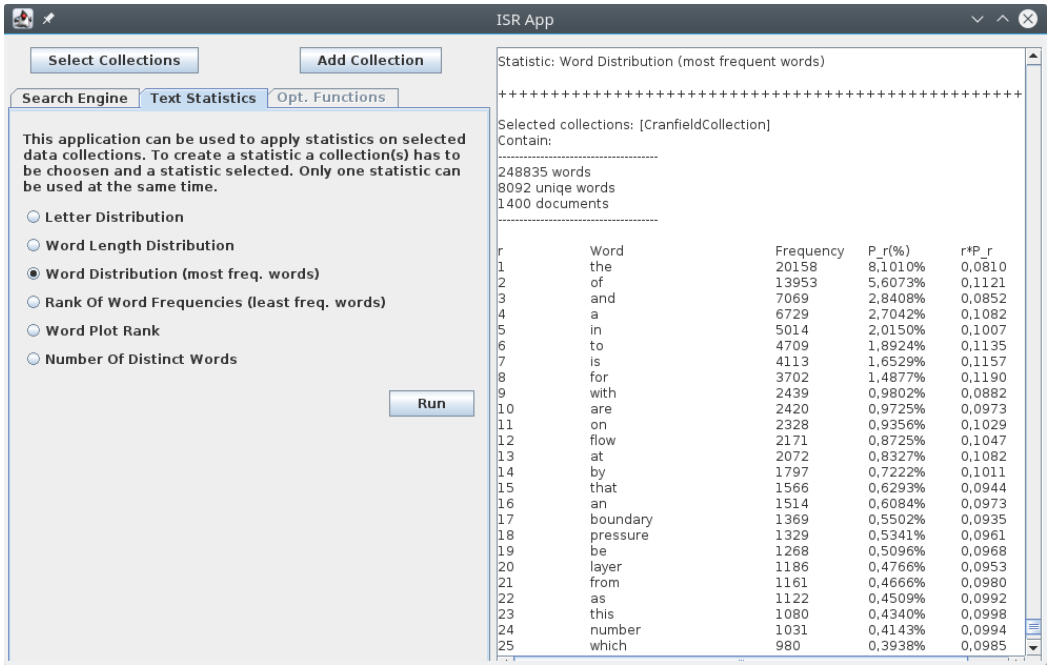

Figure 4.19: Textual output of the statistic "Word Distribution (most frequent words)" using the Cranfield document collection.

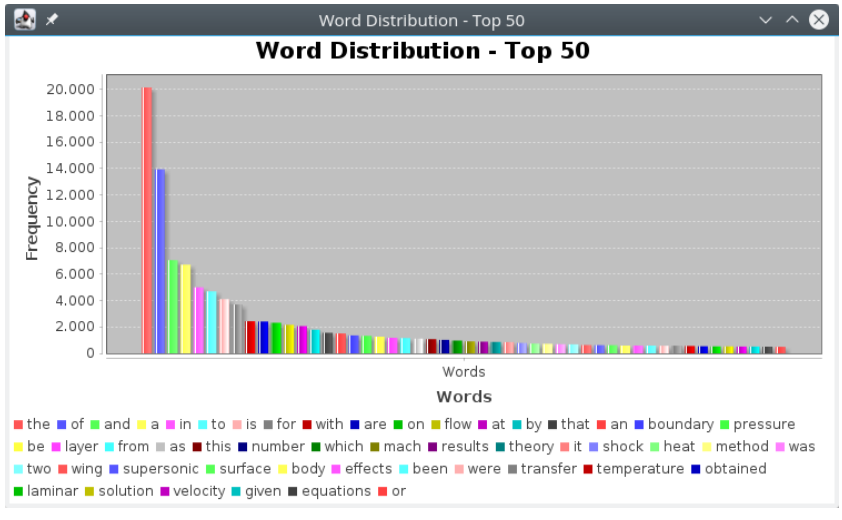

Figure 4.20: Visualization of the statistic "Word Distribution (most frequent words)" using the Cranfield document collection.

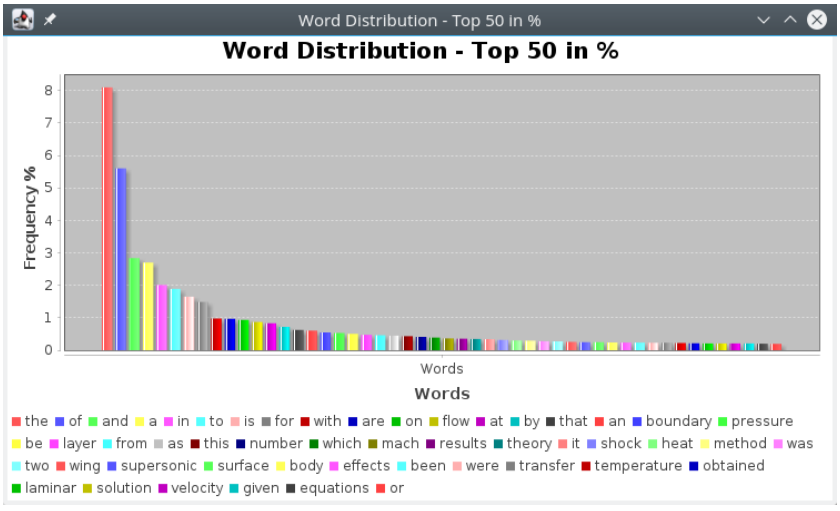

Figure 4.21: Visualization of the statistic "Word Distribution (most frequent words)" using the Cranfield document collection in percent.

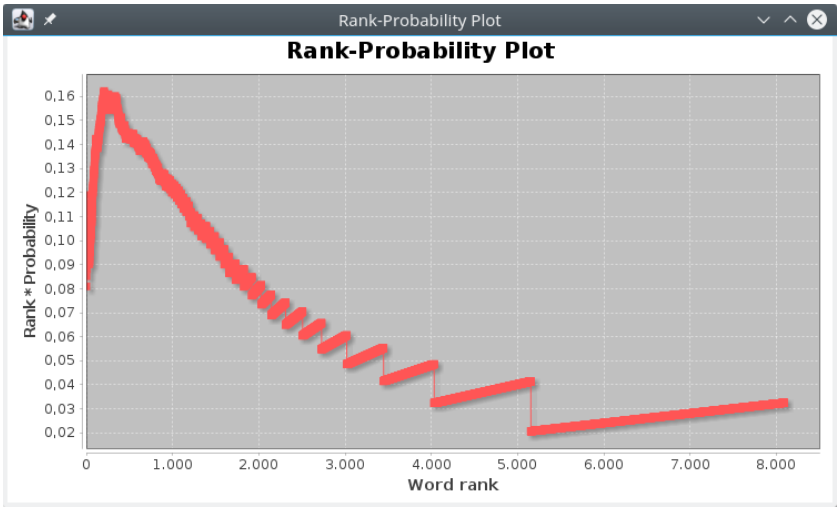

Figure 4.22: Visualization of the relation of the rank and the probability using the Cranfield document collection.

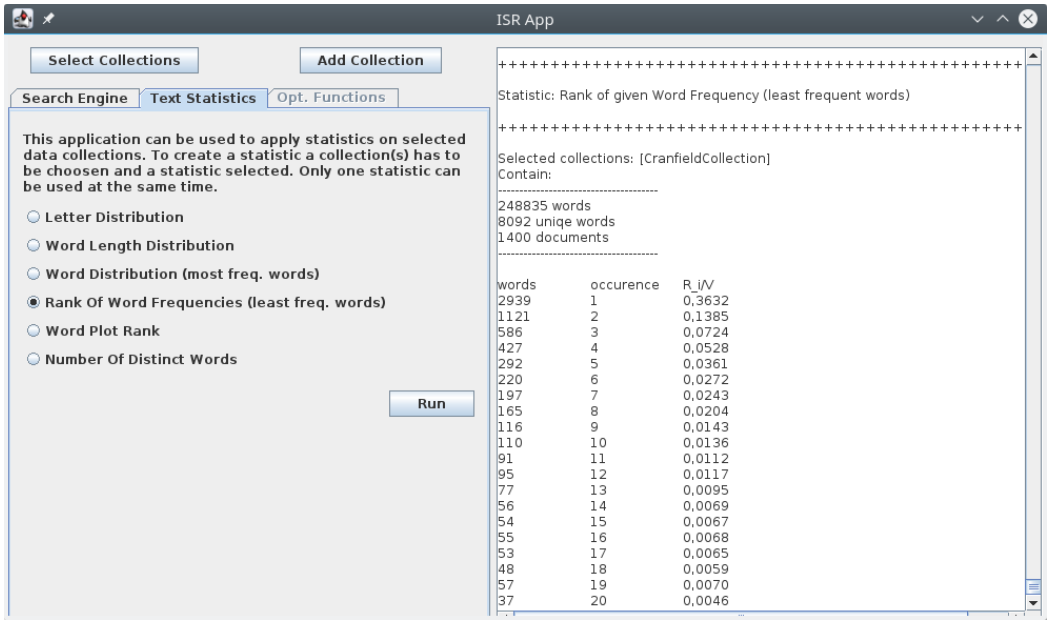

Figure 4.23: Textual output of the statistic "Rank Of Word Frequencies (least frequent words)" using the Cranfield document collection.

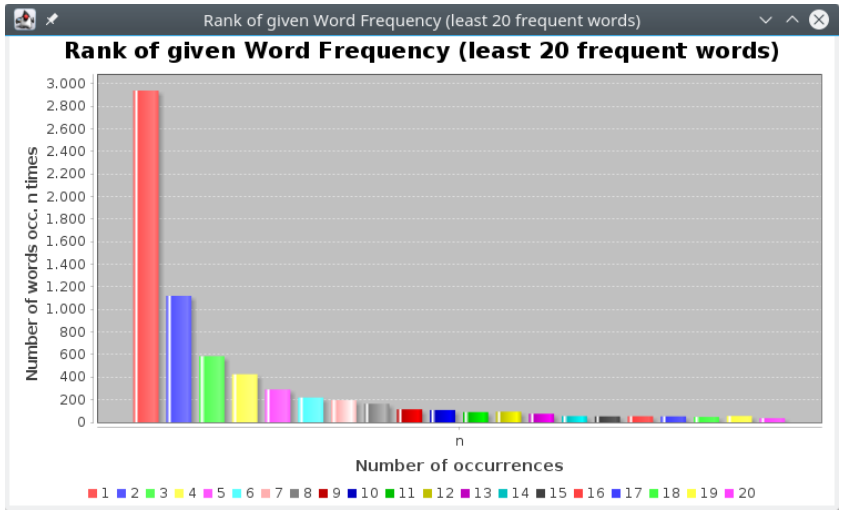

Figure 4.24: Visualization of the statistic "Rank Of Word Frequencies (least frequent words)" using the Cranfield document collection.

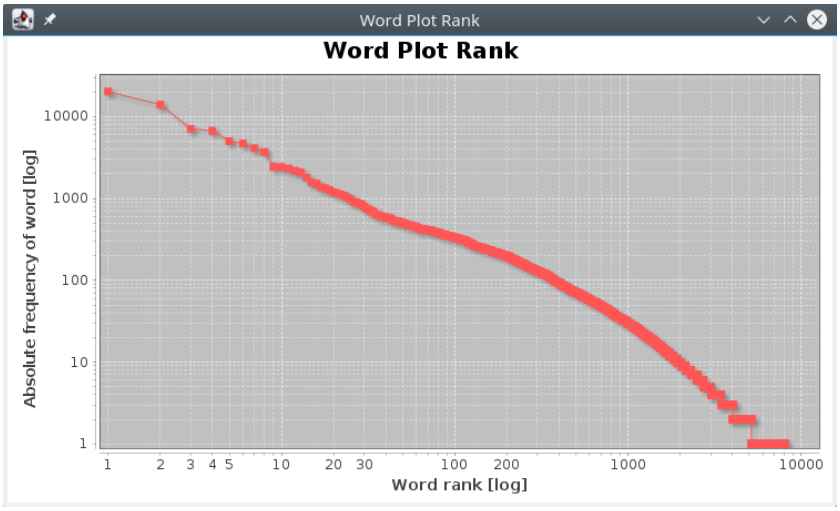

Figure 4.25: Visualization of the statistic "Word Plot Rank" using the Cranfield document collection.

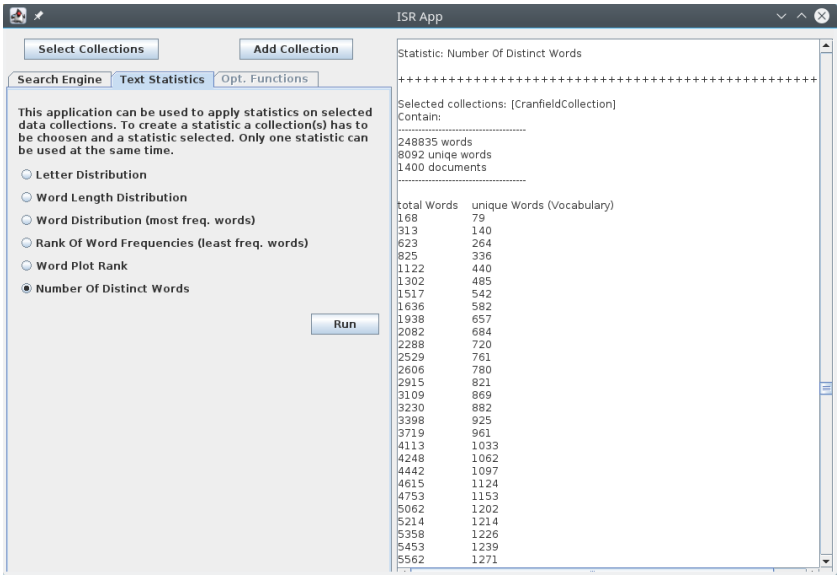

Figure 4.26: Textual output of the statistic "Number Of Distinct Words" using the Cranfield document collection.

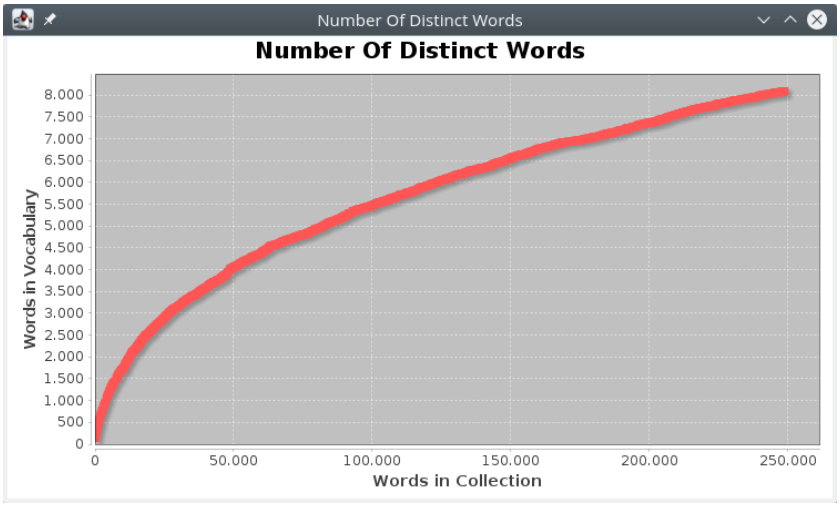

Figure 4.27: Visualization of the statistic "Number Of Distinct Words" using the Cranfield document collection.

4.5 Summary

This chapter deals with the implementation of a tool for information retrieval that provides interactivity in order to fulfill the intention of MAL (chapter 1). Based on the ideas of previous chapters, a concept for a teaching and learning tool was developed. Since the concept forces experimentations, a suitable architecture was built that supports the requirements. In order to realize the desired design, the application makes use of external libraries. Furthermore this strategy ensures also easier extension of functionalities in future development phases since it already was built in an extensible way. Based on related work from the previous chapter, findings regarding advantages and disadvantages from similar applications were taken into account in order to create a reasonable teaching and learning aid. Since nearly all reviewed tools focus on platform-independence, this was also a key factor in the creation process. Another very important aspect during the development was to feature a graphical output whenever possible.
The developed prototype ISRApp focuses in the current state mainly on two parts - one that is used to experiment with essentials of a search engine and also build it from bottom-up and another part that is used to see and experiment with characteristics of natural text. Additionally, the tool contains advanced functions which are implemented but not yet integrated in the system.

5 Evaluation and Findings

In the previous chapter, the design and development of a prototype application was covered. This chapter discusses the evaluation of the application. Results are based on user experience. The evaluation consists of two parts. Part one addresses the assembling of a search engine. The second part shows how students could work with text statistics applied on document collections. Both parts were evaluated individually due to easier analysis and interpretation of the results since the testing of the parts need a different approach. Even though both parts of the ISRApp access the same data source, each of them can be used independently. This circumstance and the fact that the ISRApp was not yet fully capable of testing also led to a qualitative and a quantitative evaluation depending on the part of the tool. However, not only the evaluation consists of two parts, also the focus in each part was different. The experiment in the first evaluation (search engine - part) targets the overall workflow of the tool while the goal of the experiment in the second evaluation (text-statistics - part) is to find out how well the user interface fits the needs of users. The first evaluation was performed by the means of printed forms, the second evaluation was conducted using the tool LimeSurvey[1]. LimeSurvey is, as the web-page claims, "the leading open-source software for online surveys". It can also be hosted at one's own server. For the evaluation, this was partly the case. The following sections of this chapter will give an overview of how the evaluation was conducted. At first, some words about the participants start the introduction. This contains both the testing group for part one the evaluation of the search engine and the subjects for part two, the evaluation of text statistics. It will give some basic information regarding their background. Afterwards, the setup and procedure will be explained followed by the results of the evaluation and a discussion of the results.

[1]LimeSurvey https://www.limesurvey.org/, accessed 2018-06-30

The second part follows the same sequence. A summary completes this chapter pointing out the relevant insights.

5.1 Participants

The ISRApp aims at students in the field of computer science. This is also due to the fact that the use of the tool requires some knowledge in programming to a certain extent regarding the part the student is experimenting with.

5.1.1 Participants for Evaluation of Part 1 - Assembling a Search Engine

To build a search engine at some point some, programming has to be done. Therefore only students with knowledge in programming were invited to participate. The recruitment was realized verbally and involved a random selection of students at the university, mainly the surroundings of the Faculty of Computer Science and Biomedical Engineering, with an interest in this field. This led to five male volunteers who were asked to experiment with the tool and build their own search engine after a brief introduction. Volunteers for this quantitative evaluation were between the age of 28 and 32 years with an average of 30 years. The evaluation of the search engine part was conducted with students who studied computer science, software development or computer engineering. Some of them also completed the course “Information Search and Retrieval” or work in a field related to information retrieval.

5.1.2 Participants for Evaluation of Part 2 - Experimenting with Text-Statistics

Interacting and experimenting with text statistics does not require previous knowledge in programming. Based on that, the evaluation of text-statistics was easy to conduct, not time consuming and could be performed during

the lecture "Information Search and Retrieval" in 2016/17, for which the tool was developed for. This included 27 male students with background in computer science. The field of study was dominated by pure computer science students but also complemented by students of software development. All participants of the classroom test were in the age between 22 and 32 years while the average was some 25 years.

5.2 Structure of the Survey

The structure of the survey is divided into three parts. A general part, that covers some very basic information about the testees (table 5.1) and general questions regarding practical experience in education also related to this tool (table 5.2). Part two is the System Usability Scale (SUS), a standard questionnaire for the appraisal of perceived user-friendliness (Lewis & Sauro, 2017). It covers the usage as well as experience after working with the tool. The questions were taken from Lewis and Sauro (2009) and are shown in table 5.3. Additionally a questionnaire regarding motivational and learning aspects during the use is applied. This refers to the Computer Emotion Scale (Kay & Loverock, 2008). The questions can be found in table 5.4. The last part concerns overall information of the application (table 5.5). This is also very important for prospective development and improvement. It may also provide ideas on how to extend the usage to other fields. The same survey was used for both evaluations.

Gender
Age
Field of Study

Table 5.1: General information of the subject.

Rate the following statements between "I strongly disagree (1)" and "I strongly agree (5)"
I like experiments and hands-on activities.
Generally, experiments and hands-on activities help me to better understand theory and methods.
This tool has helped me to better understand and reflect aspects on search engines.
I would like to use such a tool for other subjects in ISR*.

Table 5.2: Questions about practical experience in education. (*ISR - Information Search and Retrieval is the course the tool was developed for.)

5.3 Evaluation Part 1 - Assembling a Search Engine

This section will discuss the setup and evaluation procedure for the first part of ISRApp. Assembling a search engine requires programming knowledge to some extent but this evaluation focuses rather on the user experience with the tool than on implementing.

5.3.1 Setup and Procedure

The testing environment consisted of a simple office notebook with a Linux operating system. Also all necessary libraries as well as the Java runtime environment were installed to make the testing ready to start. For each testee, a total time of 45 minutes was scheduled of which 10 minutes where used for the introduction. The subject found the application in a prepared state. At that time, a brief introduction to the tool was given to the user in order provide an overview of the total range of functions. This also included the text statistics. Followed by an explanation of the tool, the user was presented the sequence of how the retrieval process works. This comprised a demonstration using the Cranfield document collection so one could see an actual search- and retrieval process. In order to test the crawler functionality in a proper way, a simple web-page was created. This contained a small number of different words and no complex structure so parsing led to a predictable result. Afterwards the subjects got some time

Rate the following statements between "I strongly disagree (1)" and "I strongly agree (5)"
I think that I would like to use this system frequently.
I found the system unnecessarily complex.
I thought the system was easy to use.
I think that I would need the support of a technical person to be able to use this system.
I found the various functions in this system were well integrated.
I thought there was too much inconsistency in this system.
I would imagine that most people would learn to use this system very quickly.
I found the system very cumbersome to use.
I felt very confident using the system.
I needed to learn a lot of things before I could get going with this system.

Table 5.3: SUS - System Usability Scale which covers questions about tool experience. Taken from Lewis and Sauro (2009)

to experiment with the tool and to familiarize their self with the workflow. Based on that, the users were given three tasks. Those included:

- Getting familiar with module locations (10 minutes)
- Crawling a web-page (10 minutes)
- Search the web-page with a predefined query using (15 minutes):
  - Boolean Model
  - Vector Space Model
  - Probabilistic Model

This procedure and testing scheme was repeated for every subject. The evaluation of the first part was completed with the survey which consists of three segments as depicted in section5.2.

5.3.2 Results

Due to a significant smaller number of participants the results reflect a quantitative evaluation and are more an indicator on how to proceed with the development. The results are split into three parts similar to the survey.

Rate the following statements between "None of the time (1)" and "All of the time (4)"
Using this software, I feel ... [Satisfied]
Using this software, I feel ... [Disheartened]
Using this software, I feel ... [Anxious]
Using this software, I feel ... [Irritable]
Using this software, I feel ... [Excited]
Using this software, I feel ... [Dispirited]
Using this software, I feel ... [Insecure]
Using this software, I feel ... [Frustrated]
Using this software, I feel ... [Curious]
Using this software, I feel ... [Helpless]
Using this software, I feel ... [Nervous]
Using this software, I feel ... [Angry]

Table 5.4: Computer Emotion Scale taken from Kay and Loverock (2008). It should answer the psychological condition during usage of the tool.

For which other subjects could such an approach be used?
What I liked most:
What I did not like at all:
What could be improved:

Table 5.5: General information regarding the tool.

5.3.2.1 Practical Experience in Education

In a range from 1-5 ("I strongly disagree (1)" and "I strongly agree (5)") all of five volunteers liked experiments and hands-on activities. This also applied to the fact that hands-on activities helped participants better to understand theory and methods. Already 4/5 fully agreed that the tool helped them to better understand and reflect aspects on search engines. Less participants (3/5) wanted to use such a tool for other subjects in ISR while none of them disagreed at all.

5.3.2.2 Experience with the Tool

In a range from 1-5 (“I strongly disagree (1)” and “I strongly agree (5)”) nobody agreed nor disagreed to the statement of if they would like to use the system frequently. Four persons did not find the system unnecessarily complex while one did not at all. This corresponds with the statement that testees expected an easy to use system. One participant out of four stated that support of a technical person would have been needed to be able to use the system. Apart from one, all agreed or even strongly agreed to the statement: “I found the various functions in this system were well integrated”. That in reverse matches the question about consistency in the system. The participants agreed or strongly agreed that most people would learn to use this system very quickly. Regarding laboriousness of working with the system, the result is indecisive. This result also applies for confidence of using the system as well as the necessity of previous knowledge in order to work with the application. Figure 5.1 shows a visualization of the key statements.

When it comes to the psychological conditions of the participants during the testing the results are more clear. In a range between 1-4 (“None of the time (1)” and “All of the time (4)”) regarding the question how users felt using the system about half of the users stated that they where neither satisfied nor exited all of the time or none of the time. This applied on average also to curiosity. All of the participants noted that they felt none of the time disheartened, anxious, irritable, dispirited, insecure, frustrated, helpless, nervous or angry. Figure 5.2 shows an overview of the emotional state.

5.3.2.3 General Experience and additional Information

All participants could imagine to use such an approach for other subjects. Mentioned topics were: “signal processing/compression methods”, “recommender systems”, “parsing mails for special content”, “data analysis”. When it came to the question what was liked most, testees answered: “easy to use”, “output for small collections like matrices”, “cool crawler for testing data sets”, “the crawler”, “speed”. Answers to to question of what they did not like at all was reflected in one reply: “design of the

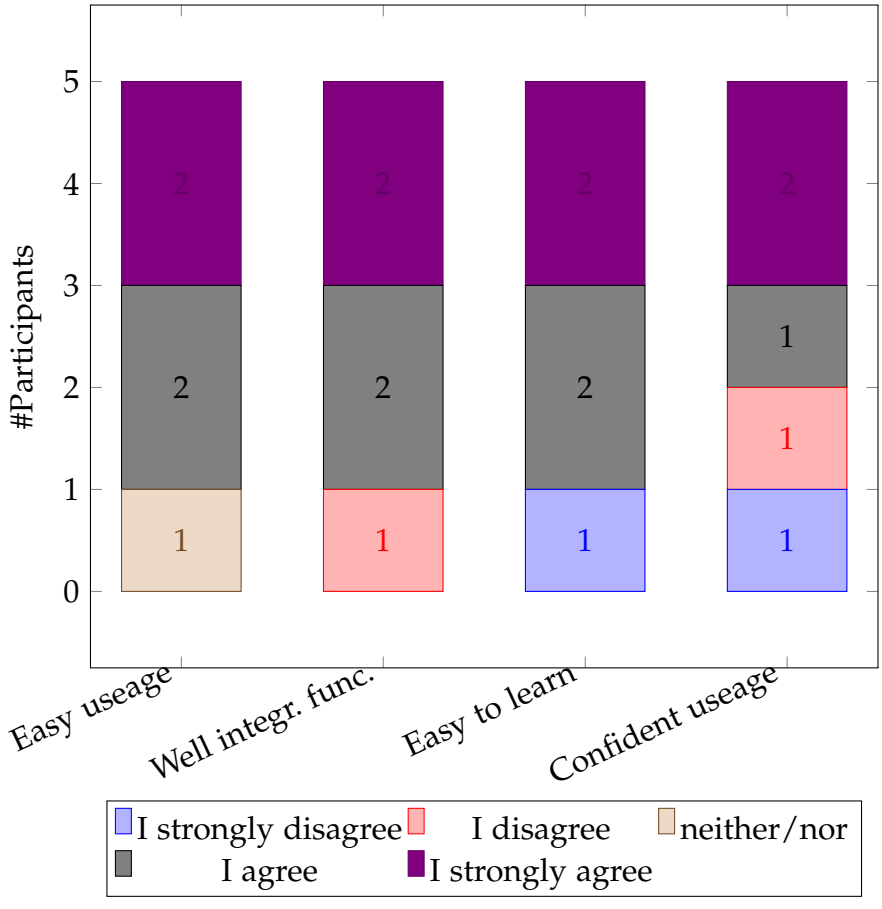

Figure 5.1: Summary of some statements from the System Usability Scale (Lewis & Sauro, 2009) after the testing of the search engine. The columns show answers regarding the statements: "I thought the system was easy to use", "I found the various functions in this system were well integrated", "I would imagine that most people would learn to use this system very quickly" and "I felt very confident using the system".

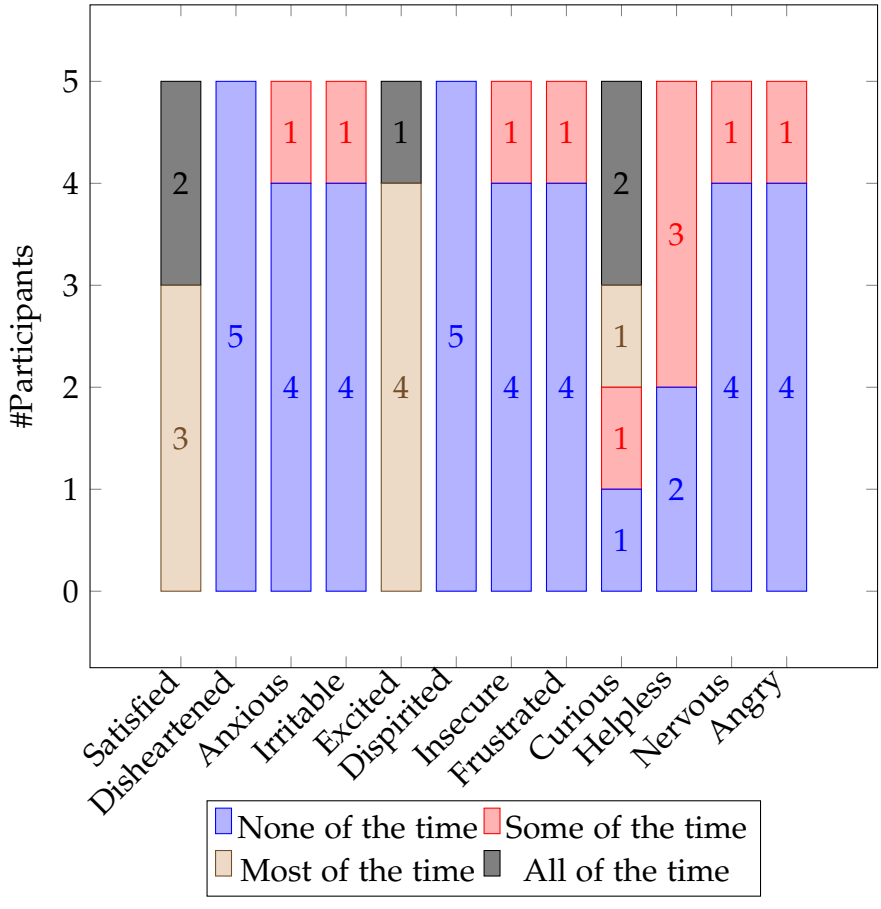

Figure 5.2: Evaluation of the emotional state during the testing of the search engine using the Computer Emotion Scale (Kay & Loverock, 2008). The plot shows how students felt using the software.

app". This also corresponds with the question on improvement. While all participants point out that the GUI needs improvement, three of them mentioned details which include: "descriptions for different search modes", "a loading bar 0-100%, AND/OR only for boolean-model", "output". There was also room for additional comments to provide information about strange behavior or hints. These included: "additional information (tooltip) for the different models", "tooltips good, but design not so responsive – large collection may keep the app running without notification to user", "bottom scrollbar of text output disappears sometimes"

5.3.2.4 Discussion

The main objective of the experiment in the evaluation of the search engine part was to get an overall impression of how intuitive the workflow is. Since the experiment was conducted with a rather small number of participants, the results may not be that meaningful. Nevertheless, the outcome already gives some important insights of the current state and for future development. As nearly all students could cope with the workflow after some minutes of introduction, there appeared situations which were not so self-explaining and required assistance. These, for example, occurred when the users had to apply different search models which assume different query formulations. Based on that users fault the absence of sufficient description even though tool tips for retrieval models are existing. It may also be a matter of background since some participants found the application easy to use. A topic that was mentioned several times was visualizations. It seemed that those were missed mainly to show the current progress. In terms of usability it is not surprising that users complain about inconsistencies regarding the state of development, but beyond that, the feedback was quite positive.

5.4 Evaluation Part 2 - Experimenting with Text-Statistics

This evaluation concerned the work with text-statistics. Compared to the evaluation in part one, this one is much more expressive since it was

conducted during a class of the course it was designed for. The testing was also easier because it did not need detailed explanations. This is due to predefined functions which simply have to be called and do not require any assembling.

5.4.1 Setup and Procedure

For this evaluation the search-engine part was deactivated so students could focus solely on the text-statistics. The exercise in this evaluation was an online task which had to be solved individually. As it was also a part of "Information Search and Retrieval" it was possible to earn points for the course. The participation was mandatory. In order to complete the entire exercise students were not bound to the time of the lecture but could submit the results within eight days. Everyone of the 27 students had to use one's own notebook. In order to provide an easier start, a simple manual was provided to guide the students through the first steps. The manual contained steps for installation and how to execute the tool. Also, important files and their functions were explained. This was concluded by the presentation of how to run a statistic and how outputs do look like. Based on that, the students had to solve tasks using the tool.
The main purpose was to investigate text statistics from a preexisting document collection and compare the results with the theory that was covered in previous classes. In the testing part of the evaluation, five tasks were given. This started with an introductory exercise for an easy entry in the subject matter but the severity increased with each task. Depending on a student's enrollment number (odd or even), a certain document collection had to be chosen. For necessary calculations a spreadsheet file should be used. Alternatively the calculations could be done using programing by the means of either the Java[2] or Python[3] programming language. Either way the submission had to be complemented with a report. The tasks were specified as follows:
a) Overall statistics

- Give an overview about the document collection.

[2]Java https://www.java.com/, accessed 2018-08-30
[3]Python https://www.python.org/, accessed 2018-08-30

b) Single letter distribution

- Administer the probability distribution on the level of single characters. Provide your findings in a table. Discuss the findings and find two application scenarios for IR making use of the letter distribution.

c) Word distribution

- Investigate the word distribution of the first 40 words (rank, word count, percentage of total). Document the findings in a table. Show in a graph how "r*pr" varies.
- Apply: $f_r = f(r) = f_1 * r^{-\alpha}$ and find for the factor α the best value (least square sum error).
- Apply: $p_r = \frac{f_r}{f} = \frac{A}{r}$ and discuss the results. Find the best value for "A" (least square sum error).

d) Number of words occurring n-times

- Apply: $\frac{R_i}{V} = \frac{1}{f_i(f_i+1)}$ and compare the values with the practical findings in a table for $N = 1 \ldots 15$.

e) Number of distinct words

- Apply: $V = K * f^{\beta}$ and estimate K and β for the given results.

After the exercises the students were encouraged to participate in the survey. The results were discussed in the following section.

5.4.2 Results

As this evaluation was conducted in a real-life scenario with students studying computer science and software development, we can already gain deeper insights of how well this part of the tool was accepted as a learning aid. Furthermore these results allow better adaptations in future developments.

5.4.2.1 Practical Experience in Education

In a range between 1-5 ("I strongly disagree(1)" and "I strongly agree (5)") half of the students agreed to the statement: "I like experiments and hands-on activities." This can be seen in figure 5.3. Over 60% agreed to the statement "Generally, experiments and hands-on activities help me to better understand theory and methods." Even about 30% strongly agreed. Figure 5.4 shows this. "This tool has helped me to better understand and reflect aspects on text statistics." was confirmed by nearly 60% as can be seen in figure 5.5. Interestingly not all students did agree to the statement "Coding the text statistics function would have helped me to better understand the theory and methods.". The distribution is shown in figure 5.6.

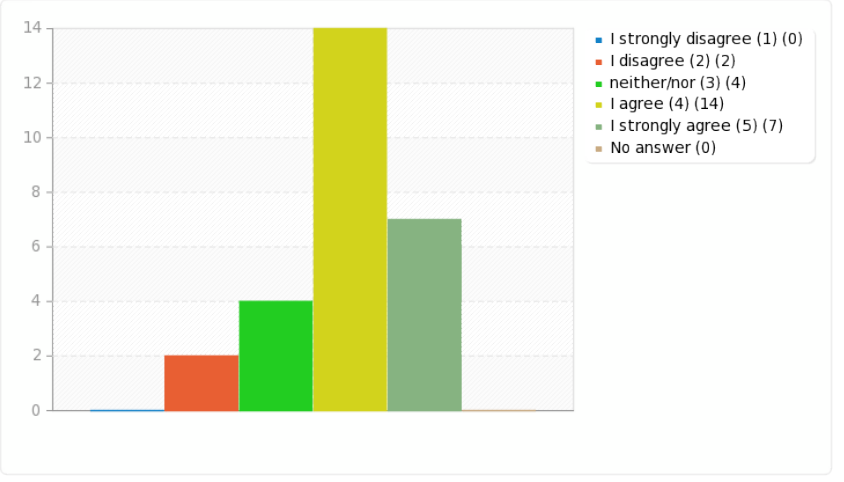

Figure 5.3: Distribution for statement "I like experiments and hands-on activities."

5.4.2.2 Experience with the Tool

In a range from 1-5 ("I strongly disagree(1)" and "I strongly agree (5)") over 40% would like to use this system frequently. In comparison to that also 30% mentioned the opposite while about 20% did not commit to a side (figure 5.7). Regarding complexity of the system, 90% did not find the tool unnecessarily complex (figure 5.8). This partially matches the statement

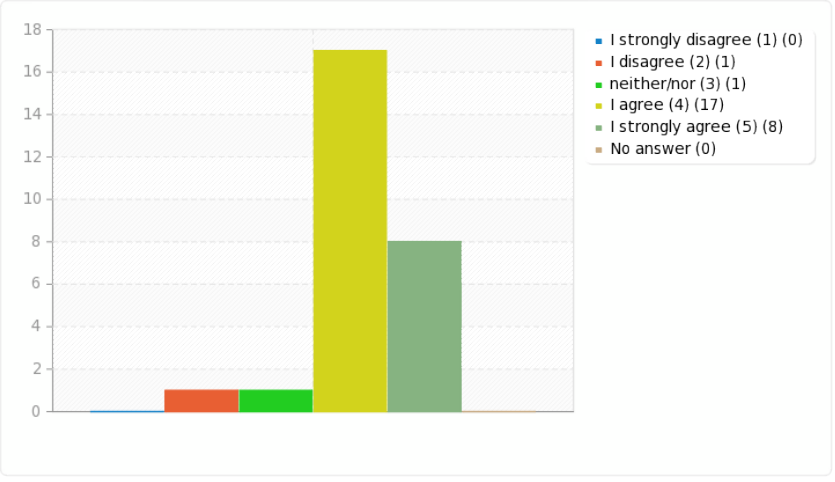

Figure 5.4: Distribution for statement "Generally, experiments and hands-on activities help me to better understand theory and methods."

"I thought the system was easy to use" where 70% agreed or strongly agreed. About 20% were unbiased. Figure 5.9 shows the distribution. Apart from one participant who did neither agree nor disagree, none of the students thought that support of a technical person was needed to use this system (figure 5.10). Most students found the various functions well integrated in the system. Only one disagreed and five neither agreed nor disagreed (figure 5.11). This result matches responses to the statement "I thought there was too much inconsistency in this system.". Distributions can be seen in figure 5.12. Also the statement "I would imagine that most people would learn to use this system very quickly" got very positive feedback. 80% agreed while 15% did not make a commitment and the rest disagreed. Figure 5.13 illustrates this. 4/27 participants found the system very cumbersome to use. Nine did not state. The remaining had an adverse opinion. In plot 5.14 the distribution can be found. The majority of participants argued that they felt very confident using the system. 20% did not make a commitment and only one disagreed (figure 5.16). Apart from a smaller group of participants who had no opinion, all disagreed to the statement "I needed to learn a lot of things before I could get going with this system". Figure 5.15 shows the result.

During the testing, participants had throughout positive impressions. The predominant majority stated that they were not disheartened, anxious

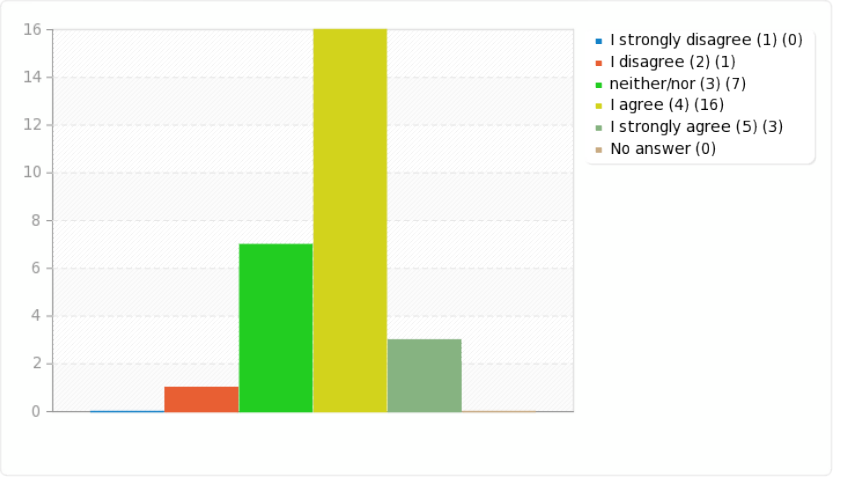

Figure 5.5: Distribution for statement "This tool has helped me to better understand and reflect aspects on text statistics."

or irritated when using the tool. Neither did most of them feel insecure or frustrated. Contrarily-working with the tool did not make students curious nor satisfied them to a large extent. The entire presentation of how participants felt using the tool is shown in plot 5.18. It can be clearly spot that the overall feedback is positive.

5.4.2.3 General Experience and additional Information

Most of the students "Would like to use such a tool for other subjects in ISR (Information Search and Retrieval - the course the tool was developed for)" as the distribution in figure 5.17 illustrates. Those participants that would like to use the tool for other subjects in ISR added examples like: "It would be interesting to analyze live data instead of test sets" or "Describing the difference between different models and its evaluation". For other subjects in general answers contained: "For the analysis of live data", "Any similar analysis", "Every other more complex concept" or "Language detection". Students liked: "the good visualizations", "It was possible to be done from home", "It was interesting to interpret the values", "Well organized and populated data, which is easy to read to use", "The usability of the tool is very high" and "The ease of use - it is really intuitive". In contrast students did not like: "reimplement the functionality given by the tool in

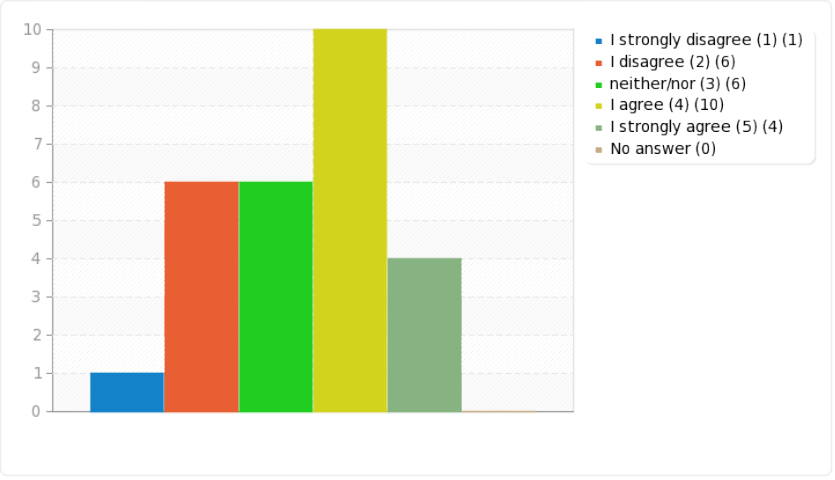

Figure 5.6: Distribution for statement "Coding the text statistics function would have helped me to better understand the theory and methods."

a spreadsheet", "not so clear instruction of the exercise", "not possible to use own collection", "parameter guessing was not that interesting", "lack of functionality and amount of logging" and "graphs do not close when other statistic is executed".
While some of the comments are related to the tool and some are just part of the way the evaluation was conducted, in other words are due to the tasks that had to be completed, there was room for improvement suggestions which are listed hereafter.
"*It would be nice to let the user decide how many results he/she wants; I would like to have the possibility to export more data or the collection used or its internal representation used; better instruction what we are asked to do, input file to analyze own texts; more clear specification regarding the tasks; When resulting in the output of data, giving a short description of the used formula and approach for what it is used; removal of inactive tabs; It might be better if the graphs would not be opened in separate windows but in tabs or something similar; more functionality and less logging*"

5.4.2.4 Discussion

The purpose of this evaluation was to find out how well users could handle the user interface that can be found in the text-statistics - tab. It turned out

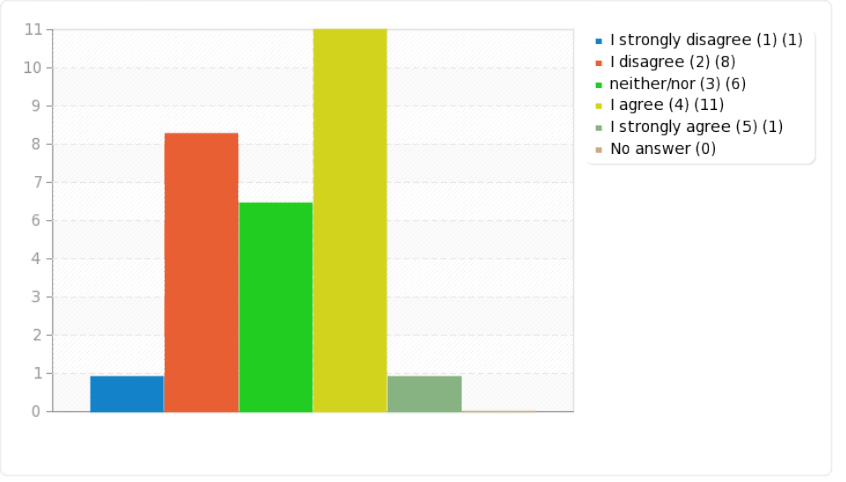

Figure 5.7: Distribution for statement "I think that I would like to use this system frequently."

that the way it was developed already conveyed a fairly mature impression. Due to a high number of participants, the results are quite expressive and can be taken seriously as a foundation in order to stabilize the existing work and extend it in further consequence. Results show that most of the users were very satisfied and felt motivated using it. The statistics-part did not show a lot of inconsistency and was easy to work with after a short introduction. All in all, participants really liked the way this part of the application was built. Especially visualizations stand out and the intuitive manner as well as a good usability seemed to be crucial for a tool users want to work with. On the downside, users complained about too much logging and the fact that visualizations do not close when other selected statistic were executed. Furthermore, the static behavior of the textual output appeared to be a problem. There might also be still some uncertainty regarding the data to be used as some user noted the lack of using own data which may be misinformation.

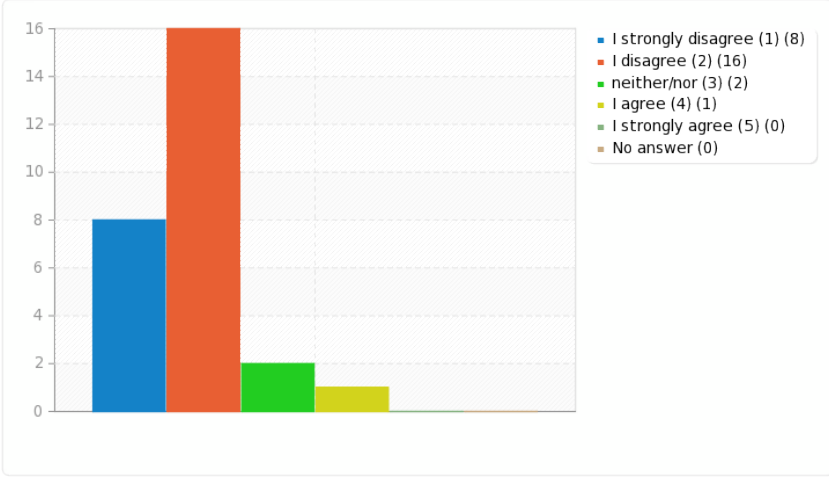

Figure 5.8: Distribution for statement "I found the system unnecessarily complex."

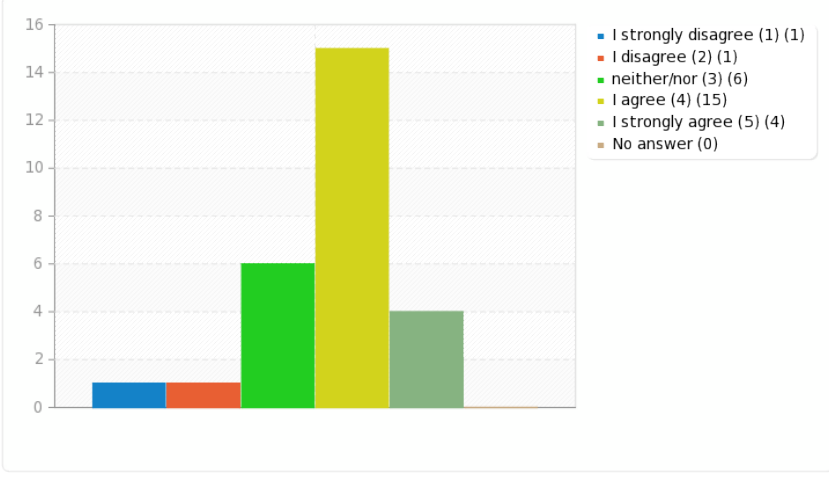

Figure 5.9: Distribution for statement "I thought the system was easy to use."

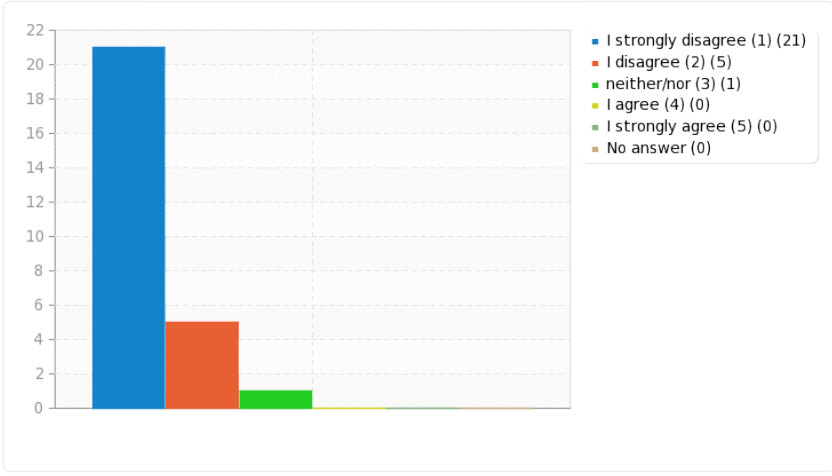

Figure 5.10: Distribution for statement "I think that I would need the support of a technical person to be able to use this system."

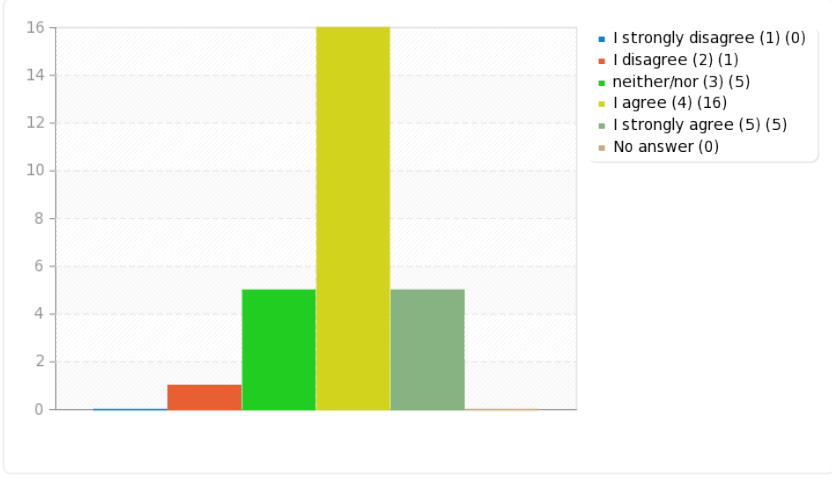

Figure 5.11: Distribution for statement "I found the various functions in this system were well integrated."

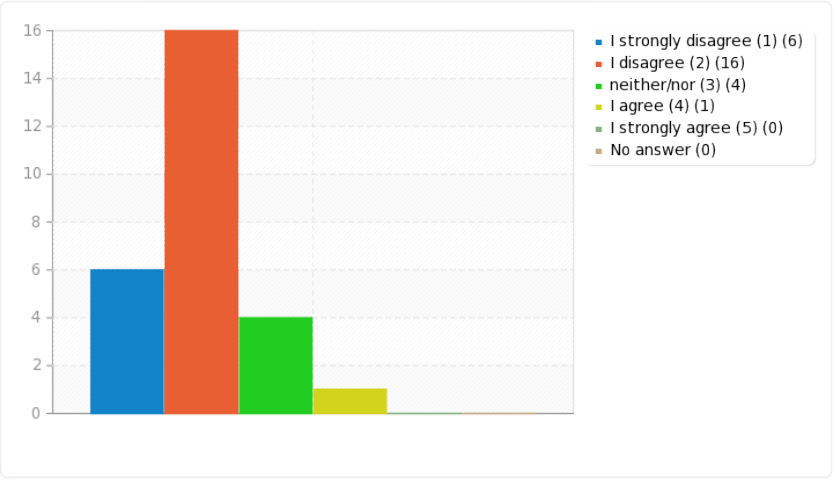

Figure 5.12: Distribution for statement "I thought there was too much inconsistency in this system."

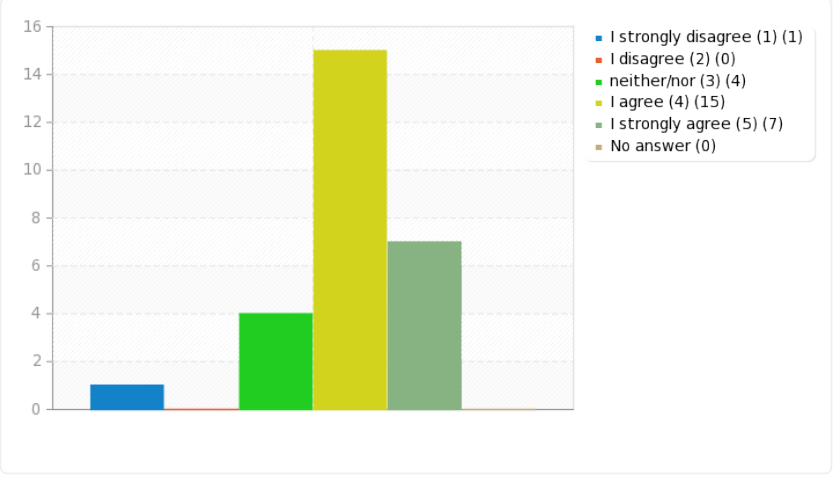

Figure 5.13: Distribution for statement "I would imagine that most people would learn to use this system very quickly."

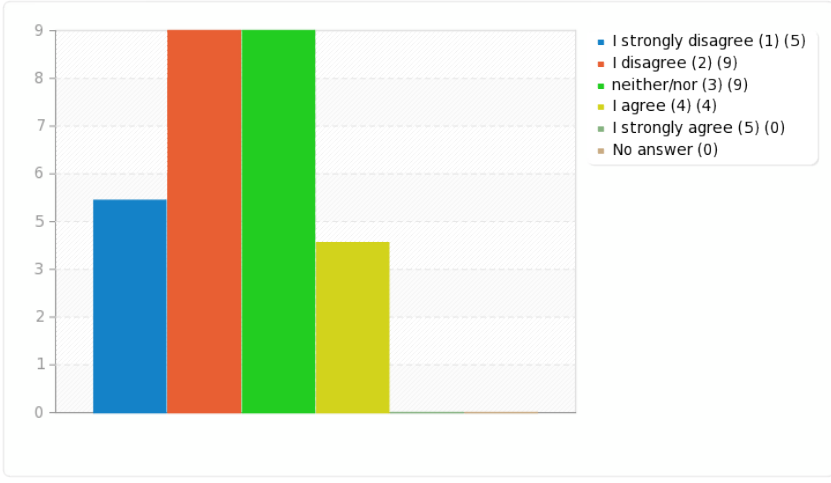

Figure 5.14: Distribution for statement "I found the system very cumbersome to use."

Figure 5.15: Distribution for statement "I needed to learn a lot of things before I could get going with this system."

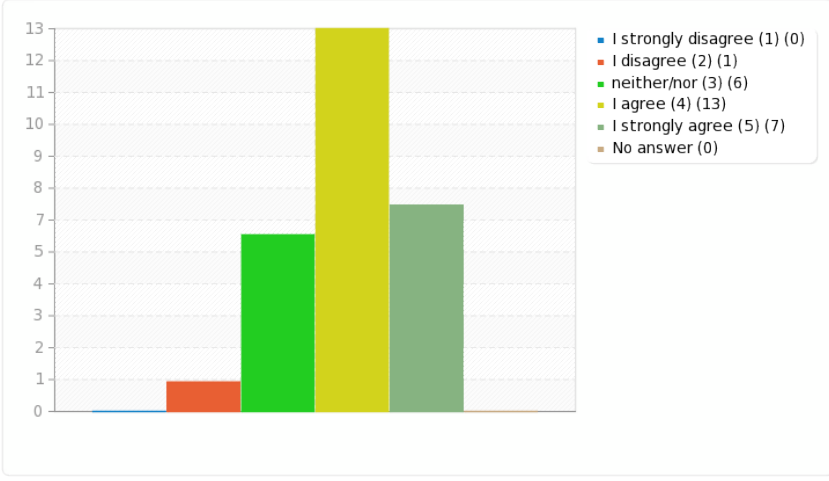

Figure 5.16: Distribution for statement "I felt very confident using the system."

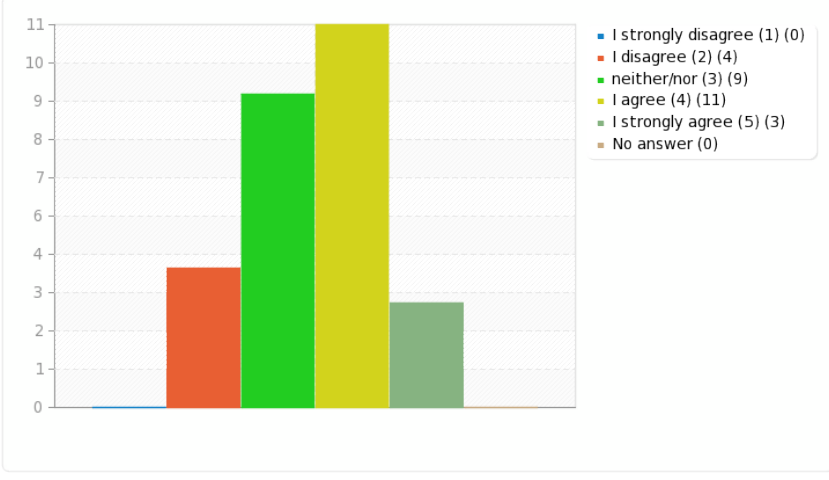

Figure 5.17: Distribution for statement "I would like to use such a tool for other subjects in ISR."

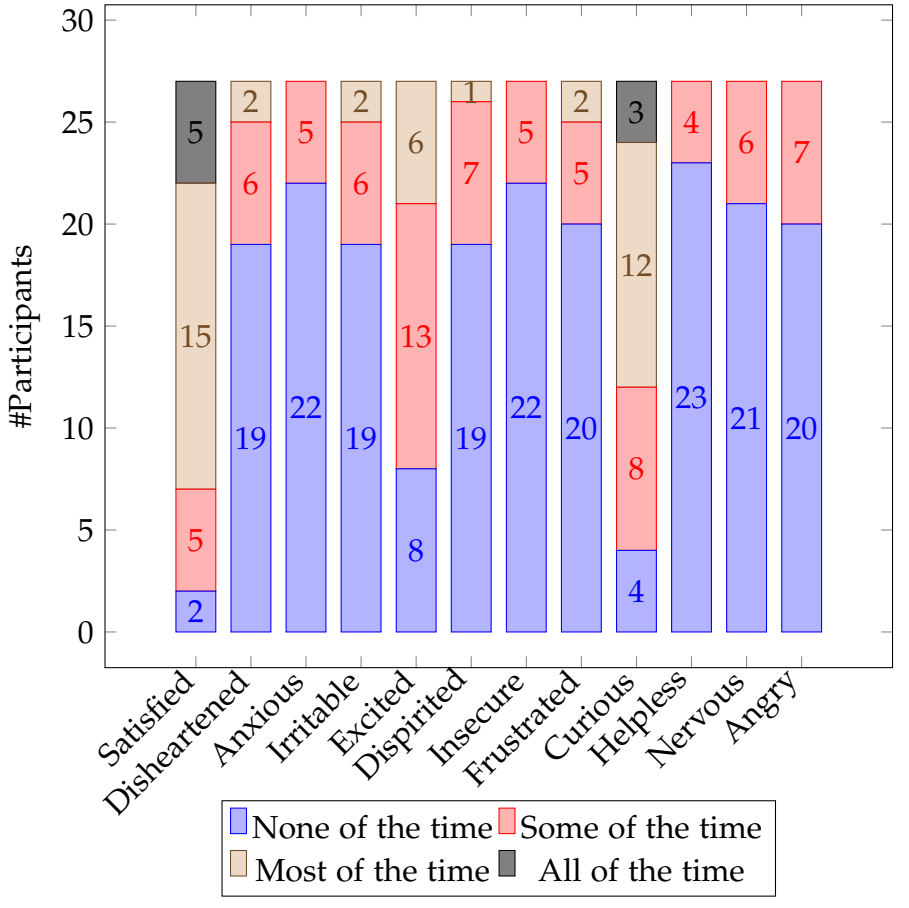

Figure 5.18: Evaluation of the emotional state during the testing of text statistics using the Computer Emotion Scale (Kay & Loverock, 2008). The plot shows how students felt using the software.

5.5 Summary

In this chapter the evaluation process of the ISRapp was elucidated. Testing and evaluation of software is essential for its success and include both a satisfied customer and great quality of software (Berntsson-Svensson & Aurum, 2006). This also applies to the ISRapp which was evaluated in two stages. This was due to the state of development of the two independent parts in the tool.
The first part of the evaluation concerned the search engine that should be built by means of the tool. Since this part was not yet ready to be tested in a real-life scenario, a quantitative evaluation was considered. This included five participants who consecutively ran through a predefined testing scheme prepared with a demonstration of the application. In part two, text statistics were evaluated by students of the lecture the ISRapp was actually developed for. This, of course, led to more detailed and meaningful results. Also the fact that all participants in both evaluation stages studied computer science or a very similar area contributed to a valuable outcome. The results of the survey showed positive feedback for both parts which also matches assumptions based on tools mentioned in related work in reference to the usage of such tools as learning aid. The way the application was developed was already well accepted by the majority of students. Experimentation as a way of gamification seemed to be a new experience for most of the participants and motivated them. Just the same as the overall concept of the tool also the usability was a main part in the evaluation process. Based on the results, nearly all students were able to handle the application after a short demonstration without problems. This is true for both parts the search engine and text statistics. The survey also confirmed that working with the tool had a positive impact for a great extent which is revealed by the emotional state of each participant. Although responses had good results, especially for the second part, there is still potential for improvement. This in particular applies to the user interface which turned out to have unexpected behavior or might at some point not be as responsive as it should be. However, this evaluation can be seen as a foundation for prospective changes and development of the ISRapp, especially regarding the part with text statistics.

6 Conclusion

E-learning in its variations is a promising alternative to traditional teaching forms as it allows location-independent and individual adapted learning experience. Nevertheless, this kind of teaching method comes along with drawbacks as discussed in chapter 2. Therefore, it makes sense to combine both approaches. The demand for e-learning is growing. This is due to the rapid development of technology and increasing requirements in education. Companies start to realize the potential of distance learning and provide courses this way. Schools and universities find e-learning as additional learning aid or supply whole lectures hereby. This applies to many different fields. When it comes to the area of STEM-education, helping tools are important to provide a support for students. The topic "information retrieval" is no exception.
This thesis takes up the concept of Motivational Active Learning MAL (chapter 1) and continues the idea by creating the possibility to experiment with fundamentals in the field of information retrieval. A first prototype was developed and evaluated in two stages which both led to positive results. The section below summarizes the major parts of this thesis. This contains a brief introduction to the problem, concepts that were found in related literature, the realization of the prototype and the results of the evaluation. In conclusion, the last section presents and discusses ideas for future development.

6.1 Summary of this Thesis

The primary goal of this thesis was to create a tool in order to implement the concept of MAL presented in chapter 1 which was applied in a course that should impart the basics of information retrieval. The concept of MAL relies on interactivity in order to increase the student's attendance through

motivational startegies with game-design connections (Pirker et al., 2014). As part of the thesis, many interesting facts came up to which attention should be paid to. Those findings and the results are in the following going to be discussed.
Foundations the developed tool - in the following also called ISRapp - are based on including e-learning, information retrieval and text statistics. The beginnings of e-learning go back to the 1960s, when the first computer based training system was developed. From that point on, many different types arose. Nowadays, there exists a broad range - from standalone systems to web-based. Moodle, for example, is a web-based system and was used for the learning format MAL. Such systems allow to track the progress of students and give feedback immediately. Unfortunately they do not provide the opportunity to experiment with the subject matter which leads to the need of an extension. The ISRapp should be the tool that fills this gap. It should have the ability to work and experiment with fundamentals of information retrieval including basic retrieval models. Furthermore, important text statistics should be covered.
As literature research revealed, some tools for the topic of informations retrieval with similar requirements were developed (chapter 3). In other areas such tools can also be found. Most of them show common functionality which resulted in positive user responses. This, for example, included a high rate of interaction with the user. Some authors focus on the integration of game elements in order to increase student's motivation. A major point is also platform-independence.
Based on related work, the prototype should implement a predefined concept picking out the positive observations of investigated tools. The concept of the ISRapp contains the ability to use local as well as online data sources. It should allow to use and experiment with both parts; foundations of a search engine and text statistics, independently. The tool was developed by the means of the Java[1] language platform. For visualizations and caching external libraries were used.
The evaluation of the prototype led to positive user feedback. It consisted of two independent evaluations, a quantitative and a qualitative evaluation. This was due to the current state of development. For the testing the Cranfield document collection was used. The results show that users could handle the application after a short introduction period without problems.

[1]Java https://www.java.com/, accessed 2018-07-10

While the search engine part still shows a lot space for improvement, experimenting with text statistics is already fairly convenient. Nevertheless, feedback revealed many weak spots mostly regarding the user interface. Apart from this, the majority of participants from both evaluations stated that they like to work with such tools. It increased their motivation. They would also like to use this kind of tools for other subjects.
In conclusion can be said that based on the feedback the development of the ISRapp already points into the right direction. Even though there is still a lot potential for improvement, parts of the tool can be used as a teaching aid.

6.2 Future Work

As this is a first prototype, it initially contains only basic functionality. According to user feedback some changes regarding the user interface have to be made. Continuing this by fixing unnecessary logging and adding additional description related to the search engine. Especially the "retrieval modes" need clarification. Users requested multiple times a more flexible output of results as well as the possibility to use own data. Both could be implemented in following versions. Apart from that, functions which are implemented but not integrated in the system will be focused on (chapter 4). Those are hidden and can be found in the third tab of the interface that is currently deactivated (chapter 4.3.4.5).
After adaptations based on user responses, thoughts about improvement of current implemented functionality can be discussed. These include the usage of libraries which improve the experimental conditions. For example Apache Lucene[2] could be applied to provide faster search for large document collections. Another feature that can be discussed is the implementation of an advanced parser in order to deal with complex web-pages. A problem we are facing when providing those advanced functionalities may be the possibly removal of the "grade-mode" which allows to compare a students implementation with a solution that is already entrenched in the retrieval field. So there is a trade-off between the state of using the tool solely for experimentation based on the provided functionality

[2]Apache Lucene https://lucene.apache.org/, accessed 2018-07-10

and the possibility to create own modules. Platform-independence is very important and due to the rise of mobile devices its worth to consider in future a transfer of the ISRapp into a web-based application. This fact was also pointed out by authors in related work.
Based on a change to web-based technology, the concept of the tool could be extended to provide the possibility to experiment in a collaborative manner. Furthermore, this would allow to implement competitions. Teams or individuals could "experiment" versus each other in order to approach to a "perfect solution" like the best search result. The time-factor could also be included.
However, there are many ideas that could make the ISRapp better and more exciting. Some of them are necessary, some do not have high priority at this point of time. It simply reveals the potential of such an application by showing just the tip of the iceberg.

Bibliography

Allahverdyan, A. E., Deng, W., & Wang, Q. A. (2013, December). Explaining zipf's law via a mental lexicon. *Phys. Rev. E*, *88*, 062804. doi:10.1103/PhysRevE.88.062804

Altmann, E. G. & Gerlach, M. (2016). Statistical laws in linguistics. In M. Degli Esposti, E. G. Altmann, & F. Pachet (Eds.), *Creativity and universality in language* (pp. 7–26). Cham: Springer International Publishing. doi:10.1007/978-3-319-24403-7_2

Baayen, R. H. (1996, December). The effects of lexical specialization on the growth curve of the vocabulary. *Comput. Linguist.* *22*(4), 455–480. Retrieved from http://dl.acm.org/citation.cfm?id=256329.256331

Baeza-Yates, R. & Ribeiro-Neto, B. (2011, February). *Modern information retrieval: the concepts and technology behind search* (2nd edition). ACM Press Books.

Baker, P., Hardie, A., & McEnery, T. (2006). *A glossary of corpus linguistics*. Edinburgh University Press.

Bakhouyi, A., Dehbi, R., Lti, M. T., & Hajoui, O. (2017, July). Evolution of standardization and interoperability on e-learning systems: an overview. In *2017 16th international conference on information technology based higher education and training (ithet)* (pp. 1–8). doi:10.1109/ITHET.2017.8067789

Berntsson-Svensson, R. & Aurum, A. (2006). Successful software project and products: an empirical investigation. In *Proceedings of the 2006 acm/ieee international symposium on empirical software engineering* (pp. 144–153). ISESE '06. Rio de Janeiro, Brazil: ACM. doi:10.1145/1159733.1159757

Bezovski, Z. & Poorani, S. (2016). The evolution of e-learning and new trends. In *Information and knowledge management* (Vol. 6, *3*, pp. 50–57). IISTE.

Bochkarev, V. V., Shevlyakova, A. V., & Solovyev, V. D. (2012). Average word length dynamics as indicator of cultural changes in society. *CoRR*, *abs/1208.6109*. Retrieved from http://arxiv.org/abs/1208.6109

Bowles, M. (2004). *What is elearning? relearning to e-learn: strategies for electronic learning and knowledge*. Melbourne University Press. Retrieved from https://search.informit.com.au/browsePublication;isbn=0522851266;res=IELHSS

Brown, S. & Fallon, C. (2002). *A guide to purchasing, developing, and deploying standards-conformant e-learning* (First Edition). CRC Press.

Bybee, R. W. (2010). What is stem education? *Science*, *329*(5995), 996–996. doi:10.1126/science.1194998. eprint: http://science.sciencemag.org/content/329/5995/996.full.pdf

Calado, P., Cardoso-Cachopo, A., & Oliveira, A. L. (2007). Ir-base: an integrated framework for the research and teaching of information retrieval technologies. In *Proceedings of the first international conference on teaching and learning of information retrieval* (pp. 2–2). TLIR'07. London, UK: BCS Learning & Development Ltd. Retrieved from http://dl.acm.org/citation.cfm?id=2228236.2228238

Cassel, J. B. (2016). Wolfram—alpha: a computational knowledge "search" engine. In N. Lee (Ed.), *Google it: total information awareness* (pp. 267–299). New York, NY: Springer New York. doi:10.1007/978-1-4939-6415-4_11

Cerulo, L. & Canfora, G. (2004, March). A taxonomy of information retrieval models and tools. *Journal of computing and information technology*, *12*, 175–194. doi:10.2498/cit.2004.03.01

Cole, J. & Foster, H. (2007). *Using moodle: teaching with the popular open source course management system*. O'Reilly Media, Inc.

Dernoncourt, F., Lee, J. Y., & Szolovits, P. (2017). Neuroner: an easy-to-use program for named-entity recognition based on neural networks. *CoRR*, *abs/1705.05487*. arXiv: 1705.05487. Retrieved from http://arxiv.org/abs/1705.05487

Efthimiadis, E. N., Fernández-Luna, J. M., Huete, J. F., & MacFarlane, A. (2011). Introduction to teaching and learning in information retrieval. In E. Efthimiadis, J. M. Fernández-Luna, J. F. Huete, & A. MacFarlane (Eds.), *Teaching and learning in information retrieval* (pp. 1–7). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-22511-6_1

Efthimiadis, E. N. & Freier, N. G. (2007). Ir-toolbox: an experiential learning tool for teaching ir. In *Proceedings of the 30th annual international acm sigir conference on research and development in information retrieval*

(pp. 914–914). SIGIR '07. Amsterdam, The Netherlands: ACM. doi:10.1145/1277741.1277982

Eklund, J., Kay, M., & Lynch, H. M. (2003). E-learning: emerging issues and key trends: a discussion paper. *Flexible Learning Framework*, *9*, 1–45. Retrieved from http://pandora.nla.gov.au/pan/39683/20040107-0000/www.flexiblelearning.net.au/research/2003/elearning250903final.pdf

Faghih, B., Azadehfar, M. R., & Katebi, S. D. (2014). User interface design for e-learning software. *CoRR*, *abs/1401.6365*. arXiv: 1401.6365. Retrieved from http://arxiv.org/abs/1401.6365

Fang, H., Wu, H., Yang, P., & Zhai, C. (2014). Virlab: a web-based virtual lab for learning and studying information retrieval models. In *Proceedings of the 37th international acm sigir conference on research & development in information retrieval* (pp. 1249–1250). SIGIR '14. Gold Coast, Queensland, Australia: ACM. doi:10.1145/2600428.2611178

Fernández-Luna, J., Huete, J., Rodríguez-Cano, J., & Rodríguez, M. (2012). Teaching and learning information retrieval based on a visual and interactive tool: sulair. In *Edulearn12 proceedings* (pp. 6634–6642). IATED.

Ferrer i Cancho, R. (2005, March). The variation of zipf's law in human language. *The European Physical Journal B - Condensed Matter and Complex Systems*, *44*(2), 249–257. doi:10.1140/epjb/e2005-00121-8

Gonzalez, H. B. & Kuenzi, J. J. (2012, August). Science, technology, engineering, and mathematics (stem) education: a primer. https://fas.org/sgp/crs/misc/R42642.pdf.

Hendry, I. G., Willett, P., & Wood, F. E. (1986). Instruct: a teaching package for experimental methods in information retrieval. part i. the users view. *Program*, *20*(3), 245–263. doi:10.1108/eb046940. eprint: https://doi.org/10.1108/eb046940

Hesse, F. W. (2009). Use and acquisition of externalized knowledge. In U. Cress, V. Dimitrova, & M. Specht (Eds.), *Learning in the synergy of multiple disciplines: 4th european conference on technology enhanced learning, ec-tel 2009 nice, france, september 29–october 2, 2009 proceedings* (pp. 5–6). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-04636-0_3

Huang, E. Y., Lin, S. W., & Huang, T. K. (2012). What type of learning style leads to online participation in the mixed-mode e-learning environment? a study of software usage instruction. *Computers &*

Education, *58*(1), 338–349. doi:https://doi.org/10.1016/j.compedu.2011.08.003

Huffman, D. A. (1952, September). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, *40*(9), 1098–1101. doi:10.1109/JRPROC.1952.273898

Ifenthaler, D. (2012). Learning management system. In N. M. Seel (Ed.), *Encyclopedia of the sciences of learning* (pp. 1925–1927). Boston, MA: Springer US. doi:10.1007/978-1-4419-1428-6_187

Jones, S. (2015, November). Plato. https://www.britannica.com/topic/PLATO-education-system. Encyclopædia Britannica, inc.

Jurjević, D., Stöcklin, N., & Hartmann, W. (2009). Informationskompetenz: ein thema für den informatikunterricht. In *Gesellschaft für informatik (gi) publishes this series in order to make available to a broad public recent findings in informatics (ie computer science and informa-tion systems), to document conferences that are organized in co-operation with gi and to publish the annual gi award dissertation.* (p. 316).

Kalimeri, M., Constantoudis, V., Papadimitriou, C., Karamanos, K., Diakonos, F. K., & Papageorgiou, H. (2014). Word-length entropies and correlations of natural language written texts. *CoRR*, *abs/1401.6224*. Retrieved from http://arxiv.org/abs/1401.6224

Kay, R. H. & Loverock, S. (2008). Assessing emotions related to learning new software: the computer emotion scale. *Computers in Human Behavior*, *24*(4), 1605–1623. Including the Special Issue: Integration of Human Factors in Networked Computing. doi:10.1016/j.chb.2007.06.002

Kolb, D. (1984). *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice Hall. Retrieved from https://academic.regis.edu/ed205/kolb.pdf%20(date%20of%20download:%2027.01.2018)

Kumar, S., Gankotiya, A. K., & Dutta, K. (2011, April). A comparative study of moodle with other e-learning systems. In *2011 3rd international conference on electronics computer technology* (Vol. 5, pp. 414–418). doi:10.1109/ICECTECH.2011.5942032

Lau, G. & Lutz, H. D. (1973). Automatic analysis of the german noun group and some problems. In *Proceedings of the 5th conference on computational linguistics - volume 2* (pp. 249–268). COLING '73. Pisa, Italy: Association for Computational Linguistics. doi:10.3115/992567.992590

Lewis, J. R. & Sauro, J. (2009). The factor structure of the system usability scale. In M. Kurosu (Ed.), *Human centered design* (pp. 94–103). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lewis, J. R. & Sauro, J. (2017, August). Revisiting the factor structure of the system usability scale. *J. Usability Studies*, *12*(4), 183–192. Retrieved from http://dl.acm.org/citation.cfm?id=3190867.3190870

Li, B. & Han, L. (2013). Distance weighted cosine similarity measure for text classification. In H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, ... X. Yao (Eds.), *Intelligent data engineering and automated learning – ideal 2013* (pp. 611–618). Berlin, Heidelberg: Springer Berlin Heidelberg.

López-García, R. & Cacheda, F. (2011). A technical approach to information retrieval pedagogy. In E. Efthimiadis, J. M. Fernández-Luna, J. F. Huete, & A. MacFarlane (Eds.), *Teaching and learning in information retrieval* (pp. 89–105). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-22511-6_7

Luhn, H. P. (1958a, October). A business intelligence system. *IBM Journal of Research and Development*, *2*(4), 314–319. doi:10.1147/rd.24.0314

Luhn, H. P. (1958b, April). The automatic creation of literature abstracts. *IBM J. Res. Dev. 2*(2), 159–165. doi:10.1147/rd.22.0159

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (first edition). Cambridge University Press.

Manning, C. & Schütze, H. (1999, May). *Foundations of statistical natural language processing*. MIT Press. Cambridge.

McLeod, S. A. (2013). Kolb - learning styles. Retrieved from www.simplypsychology.org/learning-kolb.html

Mikolajczyk, T., Moldovan, L., Chalupczak, A., & Moldovan, F. (2017). Computer aided learning process. *Procedia Engineering*, *181*, 1028–1035. 10th International Conference Interdisciplinarity in Engineering, INTER-ENG 2016, 6-7 October 2016, Tirgu Mures, Romania. doi:https://doi.org/10.1016/j.proeng.2017.02.503

Moodle. (2010). Five key principle. http://docs.moodle.org/en/five_key_principles.

Mumford, A. & Honey, P. (1992, December). Questions and answers on learning styles questionnaire. *24*.

Murphy, R. T. & Appel, L. R. (1977). Evaluation of the plato iv computer-based education system in the community college. final report.

Nadeau, D. & Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticæ Investigationes*, *30*(1), 3–26. doi:http://dx.doi.org/10.1075/li.30.1.03nad

Nichani, M. (2001, May). Lcms = lms + cms [rlos]. http://www.elearningpost.com/articles/archives/lcms_lms_cms_rlos/.

Normand, M. D., Lesmes, U., Corradini, M. G., & Peleg, M. (2010, September). Wolfram demonstrations: free interactive software for food engineering education and practice. *Food Engineering Reviews*, *2*(3), 157–167. doi:10.1007/s12393-010-9018-0

Othman, N. & Amiruddin, M. H. (2010). Different perspectives of learning styles from vark model. *Procedia - Social and Behavioral Sciences*, *7*, 652–660. International Conference on Learner Diversity 2010. doi:https://doi.org/10.1016/j.sbspro.2010.10.088

Owens, J. D. & Price, L. (2010). Is e-learning replacing the traditional lecture? *Education + Training*, *52*(2), 128–139. doi:10.1108/00400911011027725. eprint: https://doi.org/10.1108/00400911011027725

Pamfilie, R., Onete, B., Maiorescu, I., & Pleşea, D. (2012). E-learning as an alternative solution for sustainable lifelong education. *Procedia - Social and Behavioral Sciences*, *46*, 4026–4030. 4th WORLD CONFERENCE ON EDUCATIONAL SCIENCES (WCES-2012) 02-05 February 2012 Barcelona, Spain. doi:https://doi.org/10.1016/j.sbspro.2012.06.191

Pannu, M., James, A., & Bird, R. (2014, May). A comparison of information retrieval models. Richmond, BC, Canada — May 02 - 03, 2014 New York, NY, USA: ACM. doi:10.1145/2597959.2597978

Pansanato, L. T. E. & Fortes, R. P. M. (2005). Strategies for automatic lom metadata generating in a web-based cscl tool. In *Proceedings of the 11th brazilian symposium on multimedia and the web* (pp. 1–8). WebMedia '05. Pocos de Caldas - Minas Gerais, Brazil: ACM. doi:10.1145/1114223.1114231

Parsons, D. & Haden, P. (2006). Parson's programming puzzles: a fun and effective learning tool for first programming courses. In *Proceedings of the 8th australasian conference on computing education - volume 52* (pp. 157–163). ACE '06. Hobart, Australia: Australian Computer Society, Inc. Retrieved from http://dl.acm.org/citation.cfm?id=1151869.1151890

Phillips, C., Johnson, D., & Wyeth, P. (2013). Videogame reward types. In *Proceedings of the first international conference on gameful design, research,*

and applications (pp. 103–106). Gamification '13. Toronto, Ontario, Canada: ACM. doi:10.1145/2583008.2583025

Piantadosi, S. T. (2014, October). Zipf's word frequency law in natural language: a critical review and future directions. *Psychonomic Bulletin & Review*, *21*(5), 1112–1130. doi:10.3758/s13423-014-0585-6

Pillay, H. & Kappus, W. (2015). E-learning intervention for stem education: developing country case study. In L. Uden, D. Liberona, & T. Welzer (Eds.), *Learning technology for education in cloud: 4th international workshop, ltec 2015, maribor, slovenia, august 24-28, 2015, proceedings* (pp. 255–267). Cham: Springer International Publishing. doi:10.1007/978-3-319-22629-3_21

Pirker, J., Gütl, C., Belcher, J. W., & Bailey, P. H. (2013). Design and evaluation of a learner-centric immersive virtual learning environment for physics education. In A. Holzinger, M. Ziefle, M. Hitz, & M. Debevc (Eds.), *Human factors in computing and informatics: first international conference, southchi 2013, maribor, slovenia, july 1-3, 2013. proceedings* (pp. 551–561). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-39062-3_34

Pirker, J., Riffnaller-Schiefer, M., & Gütl, C. (2014). Motivational active learning: engaging university students in computer science education. In *Proceedings of the 2014 conference on innovation & technology in computer science education* (pp. 297–302). ITiCSE '14. Uppsala, Sweden: ACM. doi:10.1145/2591708.2591750

R. Jones, E. (2002). Implications of SCORM^{TM}and emerging e-learning standards on engineering education. In Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference. American Society for Engineering Education, 4–5, 8, 16, 18–19. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.8896&rep=rep1&type=pdf

Rodriguez, L. G. (2007, October). Work in progress - a dynamics course using just-in-time teaching methodology in a distance learning format. In *2007 37th annual frontiers in education conference - global engineering: knowledge without borders, opportunities without passports* (). doi:10.1109/FIE.2007.4418127

Romanelli, F., Bird, E., & Ryan, M. (2009). Learning styles: a review of theory, application, and best practices. *American Journal of Pharmaceutical Education*, *73*(1). doi:10.5688/aj730109

Salton, G., Wong, A., & Yang, C. (1975, November). A vector space model for automatic indexing. (Vol. 18, pp. 613–620). Communications of the ACM. New York, NY, USA: ACM. doi:10.1145/361219.361220

Shannon, C. E. (2001, January). A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev. 5*(1), 3–55. doi:10.1145/584091.584093

Shariat, Z., Hashemi, S. M., & Mohammadi, A. (2014). Research and compare standards of e-learning management system: a survey. *IJITCS*, *6*(2), 52–57. doi:10.5815/ijitcs.2014.02.07

Singhal, A. (2001). Modern information retrieval: a brief overview. *IEEE Data Eng. Bull. 24*(4), 35–43. Retrieved from http://singhal.info/ieee2001.pdf

Sormunen, E., Laaksonen, J., Keskustalo, H., Kekäläinen, J., Laitinen, H., Pirkola, A., & Järvelin, K. (1998). Ir game : a tool for rapid query analysis in cross-language ir experiments. In *Singapore: kent ridge digital labs* (pp. 22–32).

Stefanidis, C. & Koloniari, G. (2016). An interactive tool for teaching and learning database normalization. In *Proceedings of the 20th pan-hellenic conference on informatics* (18:1–18:4). PCI '16. Patras, Greece: ACM. doi:10.1145/3003733.3003790

Sternberg, R. & Grigorenko, E. (1997, July). Are cognitive styles still in style? *52*, 700–712.

Sun, X., Liu, X., Hu, J., & Zhu, J. (2014). Empirical studies on the nlp techniques for source code data preprocessing. In *Proceedings of the 2014 3rd international workshop on evidential assessment of software technologies* (pp. 32–39). EAST 2014. Nanjing, China: ACM. doi:10.1145/2627508.2627514

Takcı, H. & Soğukpınar, İ. (2005). Letter based text scoring method for language identification. In T. Yakhno (Ed.), *Advances in information systems* (pp. 283–290). Berlin, Heidelberg: Springer Berlin Heidelberg.

Ternier, S., Verbert, K., Parra, G., Vandeputte, B., Klerkx, J., Duval, E., ... Ochoa, X. (2009, July). The ariadne infrastructure for managing and storing metadata. *IEEE Internet Computing*, *13*(4), 18–25. doi:10.1109/MIC.2009.90

Uden, L. (2012). Human–computer interaction and learning. In N. M. Seel (Ed.), *Encyclopedia of the sciences of learning* (pp. 1464–1469). Boston, MA: Springer US. doi:10.1007/978-1-4419-1428-6_363

Wilhelm-Stein, T., Kahl, S., & Eibl, M. (2017). Teaching the information retrieval process using a web-based environment and game mechanics. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval* (pp. 1293–1296). SIGIR '17. Shinjuku, Tokyo, Japan: ACM. doi:10.1145/3077136.3084143

Yassine, S., Khalifa, M., & Franck, P. (2017, June). Towards a multidimensional model to study a critical success factors affecting continuity and success in e-learning systems. In *2017 10th international conference on developments in esystems engineering (dese)* (pp. 129–134). doi:10.1109/DeSE.2017.26

Yu, H. & Fan, J. (2009). Design and implementation of the framework for adaptive e-learning system. In F. L. Wang, J. Fong, L. Zhang, & V. S. K. Lee (Eds.), *Hybrid learning and education: second international conference, ichl 2009, macau, china, august 25-27, 2009. proceedings* (pp. 140–149). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-03697-2_14

Zaitseva, L., Bule, J., & Makarov, S. (2013). Component-based approach in learning management system development. *International Association for Development of the Information Society*.

Zhang, D., Zhao, J. L., Zhou, L., & Nunamaker, J. F., Jr. (2004, May). Can e-learning replace classroom learning? *Commun. ACM*, *47*(5), 75–79. doi:10.1145/986213.986216

Zipf, G. K. (1949). *Human behaviour and the principle of least effort*. Addison-Wesley.