The Use of Structured Text Retrieval to Search for Scientific Publications

Betreuer:
Dipl.-Ing. Dr.techn. Roman Kern

Thomas Mauerhofer (1031957)

Graz, am May 18, 2017

Contents

1 Introduction

In modern society searching for information via the Internet and specifically via Google's search engine has become a regular part of day-to-day life. While Google registered 10,000 searches a day in 1998, the same number of queries was sent per second in 2006 [1]. Therefore it stands to reason that the amount of accessible information is increasing steadily. On a daily basis, new websites are created, articles are written and scientific papers are published. To manage this amount of data and in order to distinguish between relevant and non-relevant sources various search engines are used. In this regard, searches should be simple and yet precise.

Looking at search engines for scientific publications and research there exists a broad selection of possibilities. One of the best known engines is Google Scholar. It uses a simple input interface and lists the results with respect to their relevance. These listings contain data in various formats from different years covering a variety of topics which do not necessarily reflect the search criteria. However, especially while working on a scientific paper it is vital to obtain precise results.

This paper is concerned with improving the quality of search queries and their results for scientific publications in the Portable Document Format (PDF). A common problem while searching for a specific author, for example, is that the most popular search engines often do not only list all of his or her articles and books but also sources that cite these publications. The objective of this paper is to provide measures which enhance the stability of the search terms and optimize the usability in order to deliver more concise results. To do so, simple search query structures, an intuitive front end, spell checks and post-processing in the back end will be implemented.

2 Related Work

As stated in the Introduction this intended master's thesis is concerned with improving the quality of search queries and their results for scientific publications in PDF. According to [3] all academic publications consist of similar structures. These structures are divided into chapters, sections, subsections and so on. The Structured Text Retrieval Model, as proposed in [5], can be applied to these documents very well. The model describes the handling of search queries and their corresponding results as well as the post-processing in the back end using structural meta information. In order to sort the documents with respect to their relevance, various ranking strategies such as the Jelenik-Mercer smoothing are used. Through extensions like the contextualization strategy and aggregation strategy the ranking will be refined further.

To ensure simple and stable search query structures the syntax will be constructed as described in [2]. Thereby the queries consist of multiple terms which are structured according to *label*:*keyword*. Furthermore, the terms can be prefixed with + to prioritize them specifically high. By means of this structure the user interface will be extended with an advanced search in order to improve usability and therefore create an intuitive front end.

The final usability improvement is the achieved through spell checks. These can be implemented as described in [6]. In it the Levenshtein Distance is used to verify that a word is spelled correctly.

3 Structure

At the beginning the fundamental structure for the project will be implemented. The micro framework Flask requires the front end to be coded in JavaScript and the back end in Python. Via simple requests inputs from the user side are sent to the server side. There the query is processed and a response is generated. These components form a solid ground work for the Text Retrieval Model.

Python provides a native library to connect to MySQL databases. It will be used to generate a database as proposed in [7]. This database will be filled with scientific publications via the tool described in [3]. Then a ranking system, according to [4, 5] can be implemented.

The next step is to apply spell checking to the search queries in order to enhance the usability.

Finally, an advanced search will be added that allows the user to formulate searches without having to know about the syntax of the queries, which will result in an intuitive and simple user interface.

4 Outline

1. Introduction
2. Related Work
    a) Unsupervised document structure analysis
    b) Structured Text Retrieval
    c) Structured-Text Retrieval System with an Object-Oriented Database System
    d) Search queries
    e) Real-word error detection and correction
3. Implementation
    a) Fundamental Structure
    b) Setup of the Database
    c) Userinterface
    d) Satisfaction of searchqueries
    e) Ranking system
    f) Spelling checks

4. Results

5. Conclusion

5 Selected Bibliography

References

[1] Google search statistics. http://www.internetlivestats.com/google-search-statistics/. Accessed: 2017-05-15.

[2] Sara Cohen, Jonathan Mamou, Yaron Kanza, and Yehoshua Sagiv. Xsearch: A semantic search engine for xml. In Johann Christoph Freytag, Peter C. Lockemann, Serge Abiteboul, Michael J. Carey, Patricia G. Selinger, and Andreas Heuer, editors, *VLDB*, pages 45–56. Morgan Kaufmann, 2003.

[3] Stefan Klampfl, Michael Granitzer, Kris Jack, and Roman Kern. Unsupervised document structure analysis of digital scientific articles. *Int. J. on Digital Libraries*, 14(3-4):83–99, 2014.

[4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[5] Berthier Ribeiro-Neto and Ricardo Baeza-Yates. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[6] Pratip Samanta and Bidyut B. Chaudhuri. A simple real-word error detection and correction using local word bigram and trigram. In *ROCLING*. Association for Computational Linguistics and Chinese Language Processing (ACLCLP), Taiwan, 2013.

[7] Tak W. Yan and Jurgen Annevelink. Integrating a structured-text retrieval system with an object-oriented database system. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *VLDB*, pages 740–749. Morgan Kaufmann, 1994.

6 Schedule

Step	Task in Detail	Deadline
Implementation Process	Creation of the Fundamental Structure	
	Database	
	Import of PDFs via pdf-extractor	
	Satisfaction of search queries	
	Simple Ranking System	
	Improved Ranking System	
	Implement Spell Checks	
	Improve Userinterface	
Writing Process	Introduction	
	Related Work	
	Unsupervised document structure analysis	
	Structured Text Retrieval	
	STR with an Object-Oriented Database System	
	Search queries	
	Spelling Checks	
	Implementation	
	Fundamental Structure	
	Setup of the Database	
	Userinterface	
	Satisfaction of searchqueries	
	Ranking system	
	Spelling checks	
	Results	
	Conclusion	
	Korrektur	