

Visually Linking web search results with bookmarked information

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Georg Edlinger

Registration Number 1027088

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Dr.techn. Manuela Waldner, MSc

Vienna, 30th September, 2018

Georg Edlinger

Manuela Waldner

Erklärung zur Verfassung der Arbeit

Georg Edlinger
Margaretengürtel 14/25
1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. September 2018

Georg Edlinger

Acknowledgements

I wish to express my sincere thanks to Univ.Ass. Dr.techn. Manuela Waldner, MSc for the continuous encouragement and valuable guidance. I am also grateful to Michael Mazurek for sharing his expertise with me. I also thank my girlfriend who supported me through this venture.

Abstract

This work presents a new approach of regaining access to stored information and for the visualization of similarities between new information and locally stored data. The fact that bookmarks are cumbersome to use and that there is no possibility to compare web search results with local information motivates the concept of this thesis. The implementation was done as Google Chrome extension and based on the 'Information Collage' environment. In order to improve the perceived ease of use, the visualization was integrated in the search engine results page to avoid a context switch for the user. The visualization uses a word cloud to display similarities and differences between remote and local information. The word cloud layout focuses on the spatial arrangement and the text colour of the words to encode their association to the remote or the local information.

Contents

Abstract	iv
1 Introduction	1
2 Related Work	5
3 System Overview	8
3.1 Information Collage	8
3.2 Working Pipeline	9
4 Information Retrieval	11
4.1 Keyword Extraction	11
4.2 Web Search Snippet Retrieval	12
4.3 User Snippet Retrieval	13
5 Visualization	15
5.1 Word Cloud Rendering	15
5.2 Snippet List Visualization	17
6 Implementation	19
7 Results	21
7.1 Study	21
7.2 Performance Evaluation	22
8 Conclusion and Future Work	24
List of Figures	25
Bibliography	26

Introduction

Knowledge workers, like researchers or journalists, often have to explore various topics deeply. In the process of getting familiar with a specific topic, the information retrieval model of “berrypicking” is widely applied. „Berrypicking“ refers to the approach of modifying subsequent queries based on the changing information needs of knowledge workers [Bat89]. These needs change because of their improved understanding of the domain through the selection of useful information from previous queries. Figure 1.1 shows the concept of the “berrypicking” approach.

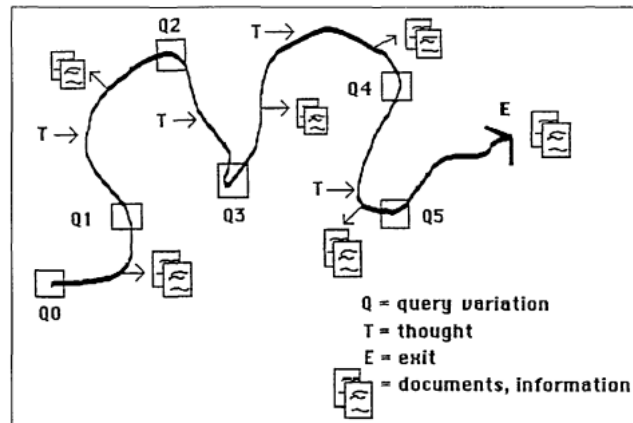


Figure 1.1: Concept of berrypicking in a information retrieval process [Bat89].

The concept of exploratory search can be understood as an extension of berrypicking. Exploratory search represents search activities that are related to learning and investigating a certain topic. Search activities can involve learning more about a topic as a step towards achieving a specific goal in an unfamiliar domain. Likewise, search activities

lacking clarity about how a goal can be achieved or without specific goals in mind are part of exploratory search. Figure 1.2 gives more detailed information about activities during exploratory search.

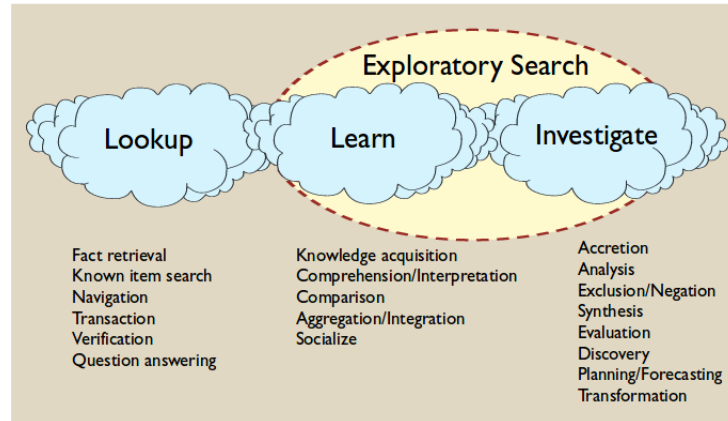


Figure 1.2: Overview of activities of exploratory search [Mar06].

The activity of exploratory search involves the processing of many different sources of information. Knowledge workers often just skim read single information sources to get an overview of the content and store relevant information for further processing. Similarly to putting various items into a physical shoebox for storing them, searched and filtered information from web resources can be put into some kind of store in order to analyse them in detail at a later time. Pirolli and Card named this store “shoebox” in their sensemaking loop model [PC05], shown in Figure 1.3.

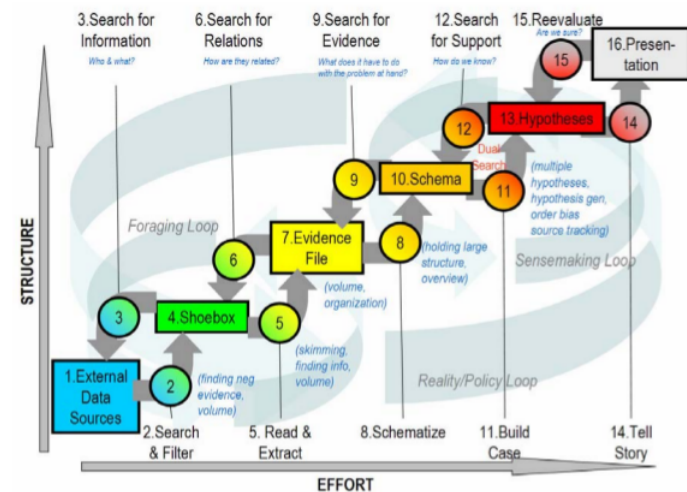


Figure 1.3: Model of sensemaking and foraging loop according to Pirolli and Card [PC05].

As the amount of stored information increases, it becomes more difficult to regain certain information from the store. Also, the determination of commonalities between already stored and new information becomes harder. The task to organize information manually in a way to support regaining and differentiation is rather difficult.

A common way to organize information from the web is to use bookmarks. The fast growth rate of bookmarks, as shown in Figure 1.4, quickly leads to the problem that they become unclear.

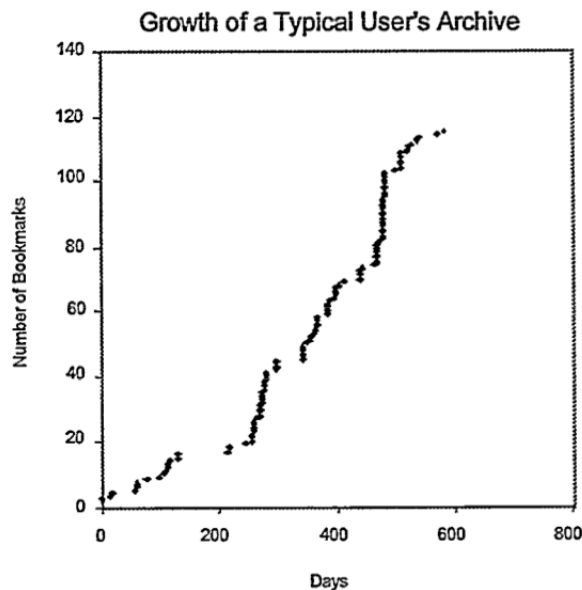


Figure 1.4: Growth rate of a typical user's bookmark archive [ABC98].

According to Abrams et al.[ABC98], people try to deal with a big amount of bookmarks by grouping and organizing them in different folders and hierarchical layers. This makeshift solution leads to the problem that a lookup has to be done in a lot of different folders and layers before a specific bookmark can be found.

Another challenge concerning bookmarks is that only the link to the full page can be stored. There is no possibility to store just a specific part of a page. Although web browsers integrate a search function in their bookmarking system, it is not possible to search only for the relevant information of the page, as the search does not consider the pages content. In order to find the relevant information again, the whole page has to be scanned, which is time-consuming.

Because of the disadvantages of bookmarks, different tools like Zotero [fHaGMU06], SenseMap [NXB⁺16] or SensePath [NXW⁺15] already try to support the organization of information. These tools can be considered a "shoebox", where information can be stored and regained again. Although the tools facilitate the retrieval of stored information, it often requires manual organization of the information into folders or by using tags.

Even if the tools support the organization of information, they lack the support of visualizing commonalities between new and stored information. Due to the rapid increase in the amount of information available on the internet, checking whether a search reveals new information compared to what has already been investigated becomes a time-consuming task, hence slowing down the searchers' workflow.

In order to counteract some of these negative effects, this thesis presents a tool for comparing stored to new information and also to re-access the stored information. The tool shows already found information related to the performed search of a user and clearly marks new information in contrast to already found one. Furthermore, visualizing similarities and differences between new and stored information should help searchers when performing exploratory search and reduces the time spent searching. While the search is performed on the web, also information snippets related to the search query get extracted from the user's locally stored data. Figure 1.5 shows the visualization which is rendered next to the search results on the search engine results page.

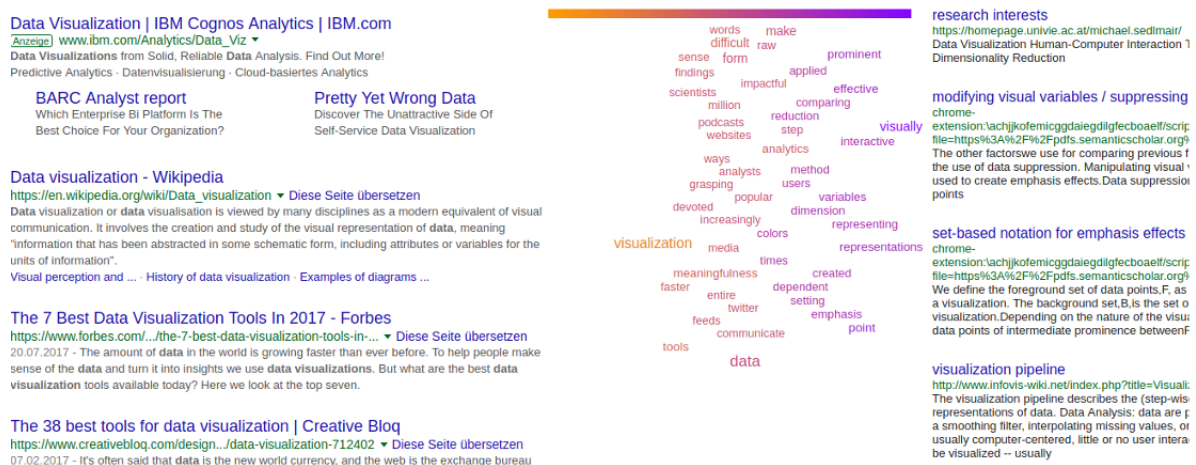


Figure 1.5: Visualization of search results.

The visualization consists of a word cloud and a list of found information snippets from the locally stored data of the user. The word cloud reveals similarities and differences between the web results and the results from the local data by rendering extracted terms of both corpora. The position on the x-axis of a term represents the association to the web corpus or the local data corpus. Terms belonging to the web corpus are on the left, terms belonging to the local data corpus on right. Shared terms are in the centre of the cloud. Besides the encoding via the position, the text colour of the term will show its association. Additionally, a list of up to ten most relevant results to the search query found in the user's locally stored data will be displayed next to the word cloud. For the evaluation of the tool, a small user study and a performance test were conducted.

Related Work

Even though tag clouds were "born" outside the world of computers [VWF09] and are often used as an emotional experience rather than an analytical tool [FFB18], there are many studies which give insight in how tag clouds could be used to analyse commonalities and differences between corpora. This section provides information about the use of tag clouds as a visualization technique in order to support activities in the field of exploratory search.

The names 'tag cloud' or 'word cloud' refer to the visualization of a document collection's content by using textual tags. These tags are associated with the frequency in which they occur in the collection [SCH08]. Historically, the terms 'tag cloud' and 'word cloud' were used to describe different concepts, but are nowadays used interchangeably. In this work, the term 'word cloud' refers to the visualization of keywords extracted from a document collection in order to summarize its content.

The aim of this thesis is to find a suited visualization method which supports users in information seeking tasks. The question arises how efficient a word cloud can be as support in information seeking tasks. This efficiency has been subject to different studies. Kuo et al. [KHGW07] showed the inefficiency of searching for a specific word in a word cloud compared to an alphabetically sorted result list. Also in Sinclair and Cardew-Hall's study participants preferred using a user interface consisting of a simple search box over the use of a word cloud to find specific information [SCH08].

Nevertheless, both studies also showed that word clouds are suitable for finding non-specific information and getting a general impression of the information content. Kuo et al. used word clouds to summarize web search results. Their findings showed that word clouds offer an overview of the knowledge represented in the response and also enables users to navigate to potentially relevant information. Another study in support of word clouds' beneficial effects on information seeking tasks was conducted by Rivadeneira et al. [RGMM07]. They found that word clouds can be useful as a means of browsing

information with no specific target item or topic in mind. According to Rivadeneira et al., word clouds can convey a general impression of the underlying data set and awareness can be raised for the most prevalent topics.

Doerk et al. [DCCW08] used the findings of Kuo et al. and Rivadeneira et al. to develop an interactive word cloud, in which they visualized a topical overview of queried data from web resources in their system. The word cloud was alphabetically sorted and the font size of the individual words represented the word’s frequency among the data items. To avoid an overcrowding of the cloud, they limited the amount of displayed words to those most frequently used. Resulting from their evaluation, the word cloud was considered a visualization tool which makes it easier to discover interesting topics, events or news.

As the effectiveness and perception of word clouds depends strongly on their layout, different studies researched the influence of individual variables. The findings of those studies were considered when designing the layout of the word cloud which is presented in this paper. Bateman et al. [BGN08] determined that font size, weight and colour of a word are most responsible for the user perception. Large words placed in the centre of the cloud attracted most user views. This was also confirmed by a study by Lohmann et al. [LZT09]. Further studies researched the influence of the spatial layout of word clouds. Felix et al. [FFB18] conducted different user studies with varying spatial layouts and value encodings. As outcome of their studies, they created guidelines for an effective word cloud design. Felix et al. also pointed out that the word cloud performance strongly depends on the task it is used for [FFB18].

Word clouds have often been designed with an emphasis on the text’s font size or its colour to encode information. The present project also makes use of the spatial arrangement of the words to encode information like Diakopoulos et al. did in their study [DES⁺15]. They developed a tool to compare word frequencies and word contexts between two corpora. Their tool used a word cloud to visualize the association of a word to two corpora. Words associated with corpus one are on the left side, words associated to corpus two are on the right side of the word cloud. Words in the centre of the cloud are shared between both corpora. In addition to the encoding via the position, the association is also encoded in the text colour of the words. The font size of a word is mapped to a word’s frequency across both corpora. In the evaluation of the CompareClouds concept Diakopoulos et al. found that the tool facilitated new kinds of comparative observations. The encoded layout helped for browsing and the visualization of the divergence score helped to filter the relevant words.

Besides Diakopoulos et al., also other studies focused on word clouds as a visualization tool to analyse corpora of documents. Castella and Sutton presented an algorithm to create a coordinated word storm in their work [CS14]. In a word storm, a set of word clouds are aligned side by side, and each word cloud represents a single document. This alignment should support the viewer in comparing and contrasting the information displayed in the individual clouds. Single word clouds were created by the idea that words that appeared in multiple documents could be placed at the same location, in the same colour, using the same orientation. Even though Castella and Sutton found that

word storms perform better than single word clouds for comparing documents, the use of multiple single word clouds still impedes a fast perception of similarities. For this task, the use of a merged word cloud like the one of the CompareClouds concept appears to be more suitable. Therefore, a word storm was excluded from the list of possible visualization methods in this work.

Similarly, Lohmann et al. focused on the analysis of multiple texts with a word cloud visualization in their concept ConcentriCloud [LHB⁺15]. They also used single word clouds to represent documents, but in contrast to the concept of word storms, the ConcentriCloud algorithm merges the single word clouds in a concentric layout. Word clouds containing words only used in single documents are on the outermost circle, whereas word clouds containing words shared among multiple documents are on the inner circles. The merging of the word clouds has the advantage that words contained in multiple documents can be found easily, because they are placed near the centre of the cloud.

Jänicke et al. presented a similar idea in their concept of TagPies [JBR⁺18]. Like Lohmann et al., they arranged the words in a merged word cloud in a circular layout. Their algorithm placed words belonging to a data category in a specific circular sector. Within these sectors, vocabulary shared by several data categories was placed in the centre, whereas words, which were unique to a single category were placed in the outer regions of the word cloud. The resulting word cloud layout can be compared to a pie chart, which facilitates the comparison of different data categories and offers an impression of their relative proportions.

To overcome shortcomings of merged word clouds like a low readability or an insufficient use of space, John et. al proposed an improved word cloud visualization, called Multi-Cloud [JML⁺18]. In order to enhance the word cloud layout, MulitCloud combines a force-based layout with a collision detection. Each document is fixed on a certain point on the border of the layout and in this way serves as a coordination system for the canvas. MultiCloud uses nodes and edges. The nodes represent words, whereas the edges stand for the relevance of a word to a document. For each word, an edge with the weighting, based on its frequency, is created. The weighting of the edges determine the movement of the word by the force-based layout. If all edge weights are equal, the word will be placed in the centre, otherwise it will be moved in the direction of the strongest weighting.

System Overview

Based on the information collage environment [Wal], a design concept was created to improve the support for exploratory search tasks. In the following sections, an overview of the existing information collage prototype will be given. Besides the overview, the working pipeline for the system will be described in detail.

3.1 Information Collage

The information collage (IC) [Wal] is a persistent information environment. It's purpose is to assist with the collection of information from the web without limitation to certain topics. The IC enables users to store text snippets or full pages. Personal notes can be used to enrich the stored information. The system displays the stored data with a screenshot of the information and additionally shows the most important keywords of the content. The keywords of a snippet are selected according to their occurrence frequency within the particular snippet and all other documents of the collage. Further explanation of the keyword selection is given in Section 4.1 The user can arrange snippets spatially on the screen. Proximate items are grouped to clusters. In the collage area, the user can vary the level of zoom resulting in changing sizes of different clusters. Figure 3.1 shows the visualization of the information collage prototype.

Similar to other personal information management tools like Zotero, the information collage has shortcomings in the way users can access stored information snippets. Accessing the stored snippets always leads to a context switch for the user. Information snippets have to be organized explicitly by the user, which is time-consuming. The goal of this thesis is to cope with these shortcomings by visualizing the most closely related information snippets of the user database to the search query on the search engine results page.

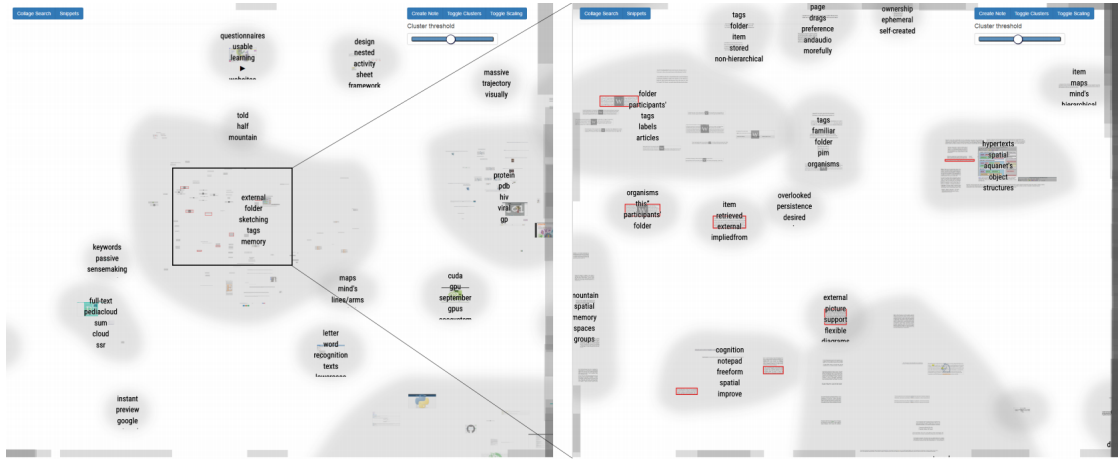


Figure 3.1: Visualization of the data in the information collage prototype [Wal].

The new design uses the prototype implementation for capturing and storing information in the browser, as well as some keyword extraction mechanisms, but implements a new way of visualizing the information snippets. As an alternative, to the collage view seen in Figure 3.1, the most relevant snippets to the search query will be displayed next to the search results on the search engine results page shown in Figure 1.5.

3.2 Working Pipeline

To enhance the already existing information collage with a better support for exploratory search, a few steps in the working pipeline have been added. An overview of steps from a user perspective can be seen in Figure 3.2a. The corresponding computational steps of the application can be seen in Figure 3.2b.

The workflow starts when a usual web search is triggered. In addition to searching the web, the application also searches the most relevant snippets for the query in the user's locally data. The output of the searches are two ranked lists, each with up to ten snippets. Every snippet consists of a title, a link, a text summary and a list with the most important keywords of the text summary. In addition to the snippets from the web search, the snippets from the user's browser database get displayed on the search engine results page. Next to the two lists of snippets, a word cloud containing the snippets' most important keywords is visualized. The position and colour of each keyword shows the association of the word to the web or to the user's local data corpus.

In order to visualize the snippets from the locally stored data and to get the keywords for the word cloud, a few computational steps are performed. The first step is to search for the most relevant snippets in the user's locally stored data. The relevance of a snippet in comparison to the search query is determined by its keywords. These keywords get extracted when the snippet is added to the stored data by the original implementation of

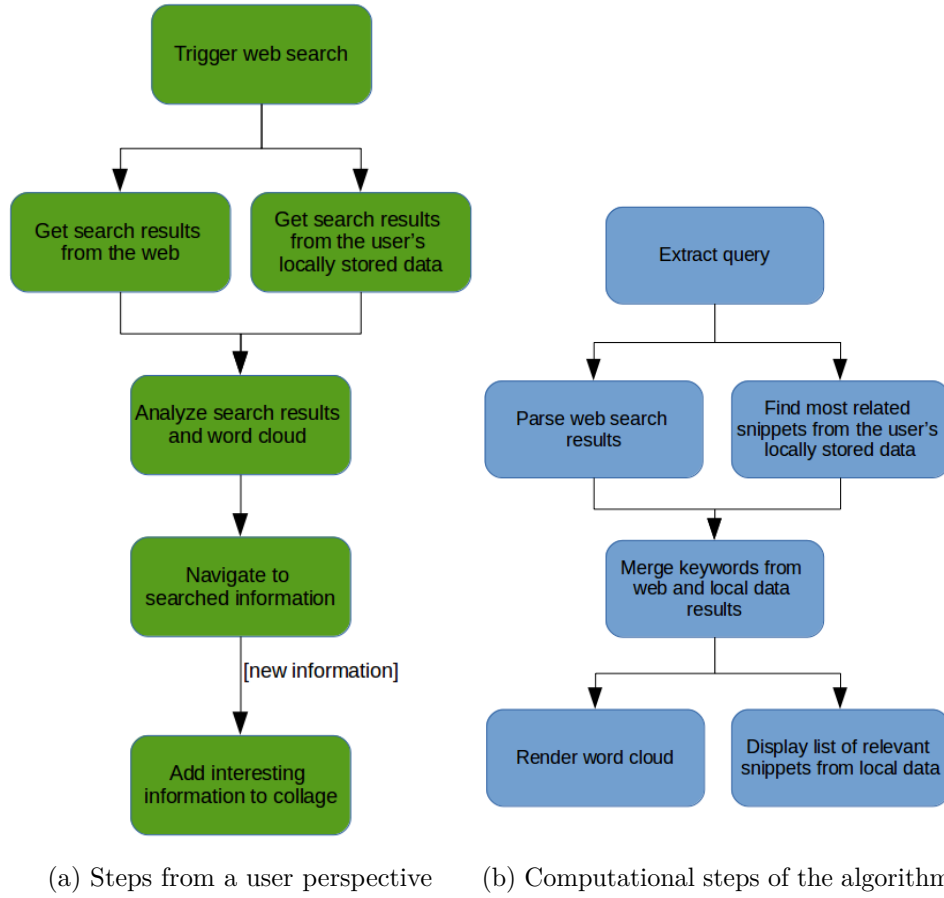


Figure 3.2: Working pipeline with user and computational steps.

the information collage prototype. Further details to the keyword selection is given in Section 4.1 and details to the selection of the most relevant snippets from the stored data is found in Section 4.3. Simultaneously with the search in the database, the results from the web search are being analysed and keywords are extracted. A detailed explanation of the processing of the web search results is provided in Section 4.2. As a next step, a list of keywords from the web search snippets and the database snippets is created. Afterwards, the association of every keyword to the web or the browser database is determined, which then influences the visualization of the word cloud. The position of every word depends on the corpus association of the keyword. After all computational steps have been completed, the results will be visualized next to the search results list on the empty display space of the search engine results page. The displayed snippets from the user's locally stored data can be used as links to get to the origin of the information.

Information Retrieval

The support of users who perform exploratory search activities requires the processing of web data and locally stored data. The visualization of differences and commonalities between the web and the local user data corpus relies on the retrieved information. The following sections will explain the snippet retrieval both from the web and from the locally stored data. Likewise, the extraction of the keywords for the snippets and the used text processing steps are described.

4.1 Keyword Extraction

In order to support the finding of similarities between the web and the locally stored snippets and to detect the most similar snippets to the query, the content of the snippets has to be in a uniform structure. Therefore, the content of each snippet is represented as a feature vector, where a feature corresponds to the weight of a single word.

4.1.1 Weighting

As a weighting scheme the term frequency – inverse document frequency (tf-idf) measure is used [SB88]. Because a feature vector characterizes a single snippet, the term frequency calculation for a word only counts the occurrences of a word within a snippet - even if the word also appears in other snippets. The term frequency $tf(t, d)$ of a word is calculated by counting the word occurrences in a snippet. The fact, that the content of snippets can vary in its length bias the value of the term frequency. Therefore, the augmented normalized term frequency (Equation 4.1) is used, which is calculated by dividing the term frequency value of each word $f_{t,d}$ by the term frequency value of the most occurring word in the snippet $f_{t',d}$ and then further normalize to a value between 0.5 and 1.

$$tf(t, d) = 0.5 + 0.5 \cdot \frac{f_{t,d}}{\max \{f_{t',d} : t' \in d\}} \quad (4.1)$$

In natural language some words appear more frequently but do not contain a high amount of information. The use of the inverse document frequency ensures that frequently used words throughout all documents are less important. The inverse document frequency $idf(t, D)$ of a word is obtained by the division of the total number of snippets N through the number of snippets containing the word, and then taking the logarithm of that quotient (Equation 4.2). The idf is calculated for one set of snippets D , called corpus:

$$idf(t, D) = \log \frac{N}{|d \in D : t \in d|}. \quad (4.2)$$

The final tf-idf score is the multiplication of the tf and the idf value, seen in Equation 4.3:

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D). \quad (4.3)$$

The words of a snippet are ranked according to their tf-idf score, and a maximum number of 50 words with the highest scores is selected as features for the vector.

4.1.2 Stemming

In natural language, words often appear in inflectional forms and not only in their root form. Since the information transmitted by the inflectional form and the root form usually does not differ in term of their contained information, only the root form of the word can be used. The benefit is the reduction of the overall number of words and the improved similarity detection of snippets. The process of reduction of inflected words to a root form by cutting of affixes is called stemming. The root form must not have the same form as the morphological root of the word. The text processing pipeline of the information collage uses the Porter stemming algorithm [Por80] for stemming.

4.1.3 Stop Words Removal

Some words of the natural language, like articles or prepositions, are very common and appear in nearly every text snippet. Their benefit in similarity detection of snippets is rather small and therefore they can be filtered out. These words are called stop words, and in the information collage they are stored as a list. They are derived from the SMART information retrieval project [BSA93] for English and from a github repository [sol16] for German.

4.2 Web Search Snippet Retrieval

In order to visualize commonalities of the search results from the web and the user's locally stored data, the content of the web search results has to be analysed and processed. Figure 4.1 shows the pipeline of the process to retrieve the snippets from the web search.

The starting point is the conduction of a usual web search by the user. The web search delivers a list of up to ten results. These results are called document surrogates and are

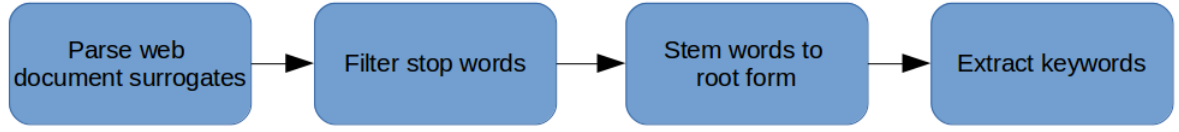


Figure 4.1: Pipeline of processing the web search results.

already ordered by the search engines algorithm according to their relevance to the query terms. Every document surrogate consists of a title, a link and a short text summary.

As soon as the search engine has finished loading the list of document surrogates, the prototype starts to process them. Every result gets parsed from the HTML of the search engine results page and will be temporarily saved as an object in an array. Every object will contain the title, the link and the text content of the document surrogate. In addition, it will contain a list of keywords which is built of the most important words of the text content. Section 4.1 explains the building of the keyword list.

Before the keyword list is created, a few text processing steps are executed to improve the keyword extraction. Filtering with a stop word list, as explained in Section 4.1.3, and stemming, described in Section 4.1.2, are executed before the keyword extraction runs.

4.3 User Snippet Retrieval

In parallel to the retrieval of the web search document surrogates, also the most relevant snippets of the browser database with respect to the query must be found. This is necessary, because the visualization displays two snippet lists: One from the web search and another from the browser database. Also for the word cloud, the list of snippets from the browser database is needed. For visualizing similarities between the web search and the database results, the keywords from the single snippets are used.

The similarity calculation requires the snippet keywords and their weights. The determination of the keywords is done when a snippet is added to the collage as explained in Section 4.1. Before the keyword calculation runs, stop words are filtered, as described in Section 4.1.3, and words are stemmed, as described in Section 4.1.2.

The similarity between a snippet from the database and the query is calculated by using the cosine similarity measure [SM86]. The cosine similarity measures the cosine of the angle between two vectors in the vector space. As the cosine is used, it means that a smaller angle between two vectors means a higher similarity between them. Each snippet is represented as a vector in the term space where dimensions correspond to all keywords of the corpus [SWY75]. A snippet vector $\vec{v}(d)$ can be expressed with $\vec{v}(d) = (t_1, t_2, \dots, t_n)$, where t_n identifies the tf-idf value [SM86] of a keyword in the snippet.

For the calculation of the cosine similarity, the query also has to be represented as a vector $\vec{v}(q)$ in the term space. Therefore, the tf-idf weights of the individual query terms

are calculated. Similarly to the processing of the user snippets, also the query terms are filtered for stop words and terms are stemmed.

The cosine similarity $\cos(\Theta)$ between a snippet and the query is defined as:

$$\cos(\Theta) = \frac{\vec{v}(d_j) \cdot \vec{v}(q)}{|\vec{v}(d_j)| \cdot |\vec{v}(q)|}. \quad (4.4)$$

Figure 4.2 shows the illustration of the cosine similarity in the term space with the dimensions 'jealous' and 'gossip' [Pre08].

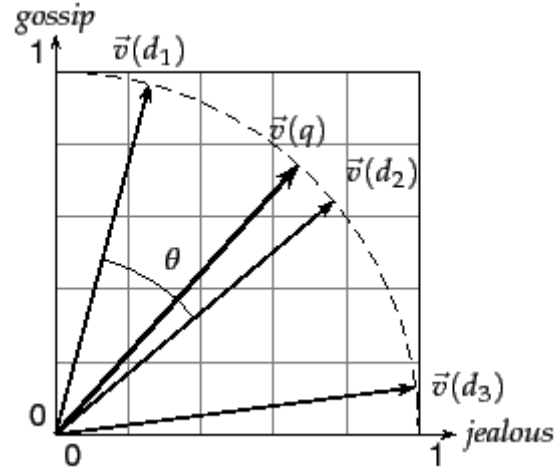


Figure 4.2: Illustration of cosine similarity between documents [Pre08].

The ranking of the snippets is done according to the calculated cosine similarity value. Snippets are counted similarly if they are at least 1.2 percent similar to the query. Based on the ranking, up to ten snippets with the highest scores are selected for the visualization.

Visualization

The two different ways of visualizing the processed data should support the user in finding similarities between the web search results and the locally stored data. It should also offer a new possibility to access already stored information. The next sections explain the visualization of the data.

5.1 Word Cloud Rendering

The possibility to visualize similarities between the web search results and the results from the browser database was a main goal of this work. Also, the support to identify possibly interesting results which are related to already seen ones was targeted. The visualization of shared words among the individual snippets from both sides, the web and the locally stored data, with a word cloud helped to reach both goals. To achieve the goals with the word cloud the layout has to meet some requirements:

1. The position of a single word should show the association of the word either to the web corpus or the database corpus.
2. The position of a single word should show the association of the word to other words from the same snippets.
3. The words are not allowed to overlap but the layout itself should still be space-efficient and minimize empty space between single words.

The retrieved snippets from the web and from the locally stored data serve as data source for the word cloud. Every snippet has a list with a different number of keywords. Some of these keywords are shared among the single snippets and appear in more than one list. For the word cloud it is undesirable to display words twice. Also the word cloud has

limitations in the maximum number of displayed words because of the limited space on the search engine results page. The solution for both is to create a list with a defined number of 100 unique keywords. The list is build by merging the keyword lists of all snippets and removing duplicate words.

For every keyword of the list the association to the web corpus or the local data corpus is calculated. The association of a term t to a corpus is determined by

$$x(t, G, C) = \frac{\sum \{g_i : t \in g_i\}}{\sum \{g_i : t \in g_i\} + \sum \{c_i : t \in c_i\}} \quad (5.1)$$

where g_i stands for a snippet from the web corpus G and c_i stands for a snippet from the local data corpus C . The value for the association ranges between 0 and 1. A keyword with the value 0 is fully associated with the web corpus and a keyword with the value 1 is fully associated with the user's locally stored data corpus.

For the visualization of the word cloud a complete graph is build where keywords represent the nodes of the graph. The graph is rendered with a force-directed layout where in a first step each node is placed along the x-axis according to its ideal position. Figure 5.1 shows the linear mapping of the nodes association to the width of the word cloud to get the ideal position of a node.

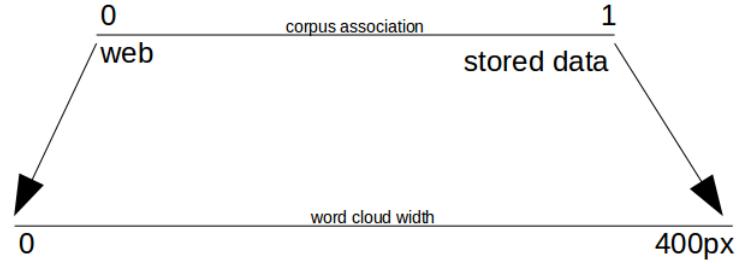


Figure 5.1: Linear Mapping of the corpus association to the x-position of a node.

The initial positioning of the words results in a layout where overlaps between words occur often and all words are shown in one line. In order to resolve such overlaps, a collision detection algorithm is used. For the collision detection, a minimum bounding rectangle with a fixed size is created for each word. The size of a rectangle is calculated by determining the rendered height and width of a word. The replacement of overlapping words is done by an iterative simulation which works with attractive and repulsive forces between nodes.

In order to meet the goals of the word cloud layout, four forces are used, three attractive and one repulsive. The defined repulsive collision force makes nodes bounce off each other upon contact of their minimum bounding rectangle. In addition, the defined attractive charge force acts as a gravity on all nodes to keep the layout of the cloud compact. Also, the attractive center force should help to keep the layout compact by pushing nodes

towards the center. The use of an attractive force to a nodes ideal position on the x-axis should help to keep the nodes association to a corpus by preventing a random placement of nodes.

In addition to the spatial arrangement also the text colour of a word encodes the association to a corpus. For the encoding, the value of the association is linear mapped to colours between orange and purple in the CIELAB colour space. Orange is associated with the web corpus and purple with the stored data corpus.

The font size of a word is determined by its normalized term frequency over all snippets from the web and from the stored data corpus. The range is from 12 pixel to 20 pixel and the transformation from the term frequency value to the pixel value is done with a linear mapping.

Figure 5.2 shows a conceptional layout and the implemented layout of the word cloud. The conceptional layout shows the invisible minimum bounding rectangles of a word and the encoding of the corpus association by the position and the text colour.

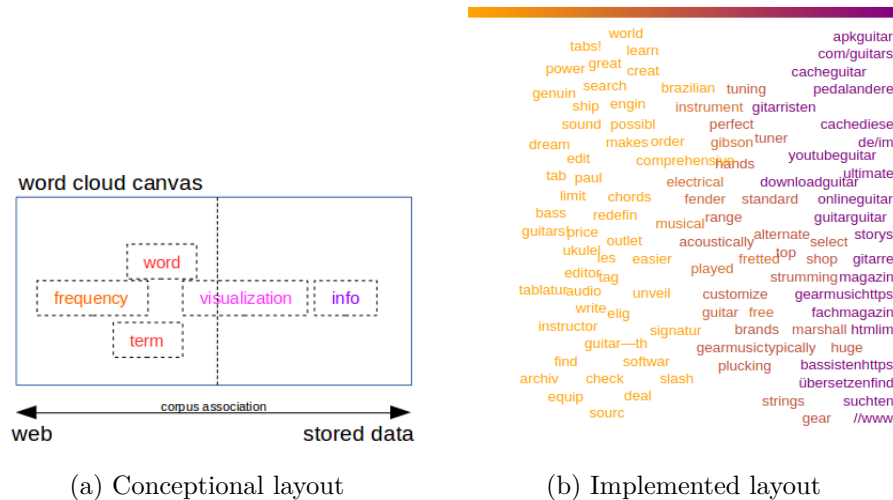


Figure 5.2: Concept and implemenation of the word cloud layout.

5.2 Snippet List Visualization

Besides the visualization of the word cloud, also the relevant snippets from the browser database should be shown on the search engine results page. This approach should support the user to distinguish between new document surrogates and those already seen. It should also help to find previously stored information easier again because the user needs only to remember a keyword or the topic of the stored information. The visualization of the snippets is based on the visualization of the document surrogates of the web search. For every snippet, the title will serve as link to the origin of the

stored information snippet. In addition, to the title, the URL to the origin and a short text summary of the stored snippet will be displayed. The text content of the single snippets may vary in its length, to ensure a uniform representation of the text summary the maximum length must be limited. If the text content exceeds the maximum length of fifty characters, the content will be cut after the fiftieths character. A click on the title of the display document surrogate navigates to the original source of the information. Figure 5.3 shows the resulting visualization of the snippet list.

parlor guitar
https://en.wikipedia.org/wiki/Parlor_guitar
Parlor or parlour guitar usually refers to a type of acoustic guitar smaller than a Size No.0 Concert Guitar by C. F. Martin & Company.

<http://www.guitaradventures.com/top-5-best-parlor-guitars-that-dont-suck>
For this reason, what has become known as the "parlor guitar" was born. I put that in quotes because, to be frank, there is no universally-accepted definition for a parlor guitar. There are certain characteristics that are common in most common parlor guitars including a smaller width, elongated lower bout and a 12th-fret

<http://usa.yamaha.com/products/musical-instruments/guitars-basses/ac-guitars/fg/fg700s/>
ENTRY LEVEL ACOUSTIC.DELUXE FEATURES. The heritage of Yamaha guitars begins with the FG line of acoustic guitars. Great entry level acoustic guitar with deluxe features including die-cast tuners, solid sitka spruce top, and a rosewood fingerboard.

<http://trustyguitar.com/beginners-guitar/small-hand-guitar/>
Martin LX1 Little Martin Martin guitars need no special introduction. This company has established its reputation a long time ago, and their instruments are simply some of the best in the class. Martin's LX1 Little Martin is a travel, also known as parlor guitar that brings you a good portion of

<http://trustyguitar.com/beginners-guitar/small-hand-guitar/>
Yamaha FG700s With Good Small Hand Reviews The Yamaha FG700S is a really great quality acoustic guitar in all regards for the price range. Easily compares to guitars priced in a much high range. There are multiple aspects of the build and wood on this instrument that give it such a

<http://trustyguitar.com/beginners-guitar/small-hand-guitar/>
Epiphone With Slim Taper Neck The Epiphone PR-5E is a slim line guitar model that has been around for years and is designed primarily for entry level players. Its quality and build, however, will please even more seasoned players who are willing to give it a chance. Despite the slim line build

Figure 5.3: Visualization of stored snippets

Implementation

The concept was implemented as a Google Chrome extension based on the already existing information collage extension and consists of two main parts. The content script, which is interacting with the Google site, and the background script, which is responsible for processing and storing the data. The communication of the two scripts is asynchronous and a concept can be seen in Figure 6.1. Both scripts are written in JavaScript.

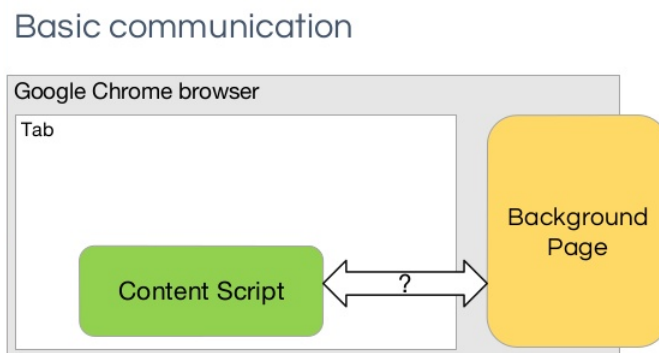


Figure 6.1: Concept of communication of content and background script [Zin15].

The content script uses JQuery to interact with the Google results page. It reads the HTML of the page for parsing the web search results and injects the word cloud and the snippet list on the page. The background script is responsible for text processing, and calculations like the similarity calculation and interacts with the data layer. As data layer, the extension uses the IndexedDB. The IndexedDB is a client side storage API which uses key-value pairs to persist data in the user's browser.

For the visualization of the word cloud, the D3 framework [Bos17] with a force directed algorithm [Rot17] was used. Because the D3 algorithm was designed to detect collisions

only between circular nodes, the `d3-bboxCollide` library [Mee17] was used to handle rectangular nodes. In order to detect overlaps between nodes, a minimum bounding rectangle had to be build before the collision detection was triggered. To determine the size of the minimum bounding rectangle, each word was rendered as SVG text element. Afterwards, the computed bounding box of the text element was used to define the top-left and bottom-right corner of the minimum bounding rectangle for the `d3-bboxCollide` algorithm. Figure 6.2 shows the code snippet to build a minimum bounding rectangle and to initialize the simulation for the force directed layout.

```
var collisionDetection = d3.bboxCollide((a) => {
  return [[- (a.width*0.55), - (a.height*0.55)], [a.width*0.55, a.height*0.55]]
})
  .strength(1)
  .iterations(2)

d3.forceSimulation(data.nodes)
  .force("collision", collisionDetection)
  .force("x", d3.forceX(a => a.originalX).strength(0.7))
  .force("charge", d3.forceManyBody().strength(10))
  .force("center", d3.forceCenter(width/2, height/2))
  .on("tick", ticked);

function ticked() {
  textNode.attr('x', function (d) {
    return d.x = Math.max(d.width, Math.min(width - d.width, d.x));
  }).attr('y', function (d) {
    return d.y = Math.max(d.height, Math.min(height - d.height, d.y));
  });
}
```

Figure 6.2: Code snippet of initializing the simulation and the collision detection.

The nodes of the word cloud were built with the data received from the background script. The background script provides an array of objects where each object represents a node of the graph. For the linear mapping of a node from the corpus association to the position on the x-axis, linear scales from the D3 framework are used. Scales are also used for the mapping of the association to the colour and for the mapping of the term frequency to the font size.

Results

The evaluation of the implementation was done with a qualitative longer-term study and a performance test. In the following sections, the methods and results of the evaluation are described.

7.1 Study

The goal of the study was to evaluate the usability and the perceived ease of use of the implementation. The study was conducted with five participants. According to Jacob Nielsen, five users are enough to find up to eighty percent of all usability problems of the application [Nie]. Each participant used his own laptop to take part in the study and installed the Google Chrome extension. Compared to a study in a usability lab [Sto02], this specific setup had the disadvantage that the hardware was very different and could possibly lead to a falsification of the results. On the other hand, it had the advantage that the visualization could be tested for different screen sizes. The participants, four male and one female student were aged between 22 and 27 years. Four of them were enrolled in computer science, one was studying veterinary medicine. The five participants had no specific task to fulfill. They got an instruction how to use the extension and were asked to use the extension within their daily life and search routines for one week. At the end of the week, they had to answer an online questionnaire. As template for the questionnaire, the System Usability Scale (SUS) was used [B⁺96]. The SUS consists of ten questions. For each question, the answer can be given on a five-point scale where 1 is ‘strongly disagree’ and five is ‘strongly agree’. The interpretation of the score follows the findings of Bangor et al. [BKM09] shown in Figure 7.1. In addition to the online questionnaire, also a short interview with every participant was conducted.

Table 7.1 shows the scores of the individual questionnaires. Four in five participants rated a high usability and perceived a high ease of use. Participant four rated a poor usability mainly because of the low functionality of the extension and the inconsistent layout of

	SUS Score	Adjective Rating
Participant 1	87.5	Excellent
Participant 2	82.5	Good
Participant 3	92.5	Excellent
Participant 4	67.5	Ok
Participant 5	75	Good

Table 7.1: SUS Scores of the questionnaires

the word cloud for the same query. In the interview, participant four said: "I would expect the same word arrangement in the cloud for searching twice with the same query terms. I was surprised, that the arrangement was different for the same query, although I did not alter the stored data". The participant also said that he would have liked more interaction possibilities with the word cloud, like clicking on words to show their origin. Also other participants found the word cloud layout was sometimes inconsistent. One participant said that sometimes the colour encoding did not fit the spatial arrangement of the word. Another participant mentioned that it was not always clear how the content of the snippets is represented by the displayed keywords.

According to all five participants, the extension is not unnecessary complex, easy to use and people would learn quickly to use it. One in five participants could imagine to use the extension frequently, especially as a support in the research activities at university. In the interview, all of them mentioned that they think the extension could be useful if the functionality was improved.

7.2 Performance Evaluation

For the performance test, the elapsed time of the individual working steps was measured and logged. The aim was to evaluate the performance of the single working steps and to find possible bottlenecks in the working pipeline. For the performance evaluation, a computer with a 2.2GHz core i3 processor was used and the speed of the internet connection at the beginning of the test was measured with 4.1 Mbps. Four single test runs with different browser database sizes for the query "information visualization" were

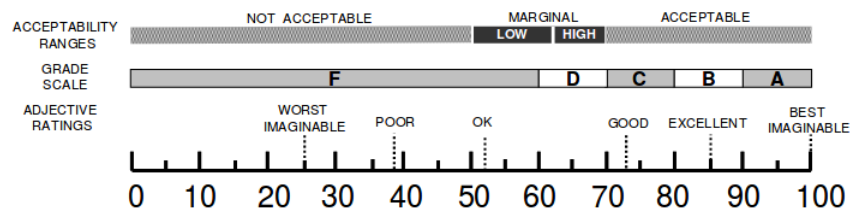


Figure 7.1: Interpretation of SUS score [BKM09].

performed. The elapsed time of the operations for the single runs can be seen in table 7.2.

Run	Web snippets	User snippets	Keyword selection	Tag cloud	Snippet list	Overall time
A	115	0	292	205	0	612
B	115	2288	367	285	115	3170
C	115	1112	326	241	0	1794
D	115	1129	345	270	118	1862
E	109	1137	342	271	115	1865
F	123	1133	346	269	116	1864

Table 7.2: Runtime of individual operations in milliseconds,

Run A was with an empty browser database. The overall time of this run was the shortest because of the missing database data. There was no need to retrieve and visualize snippets from the browser database. As a result of an empty database, the keyword selection was not that complex and the word cloud rendering was also easier and faster.

Run B was the opposite of A and a database with 52 snippets was used. The user snippet retrieval was the biggest part of the total runtime because of analyzing the high number of stored snippets. Also, the selection of keywords and rendering them in the word cloud needed more time.

The database for run C and D had a size of 26 snippets. For run C, the database served no results for the query. For run D the database served more than twenty results for the query.

The retrieval of the web snippets was over all four runs a constant time, because the search engine results were always the same. To measure the runtime of the web snippets retrieval, two additional runs were started with different queries. The search of run E used the query 'guitar tutorial' and the search of run F used the query 'facebook zuckerberg'. The runtime for run E and F was very similar to the previous runs because the results of the search engine do not differ much in their style. The length of the title or the text summary may vary a little bit.

The user snippet retrieval has the biggest influence on the overall runtime of the extension. Especially a big database slows down the algorithm significantly. Besides, also the selection of the keywords can have a bigger influence, especially if the single snippets have long lists with keywords. Minor influence has the web snippet retrieval, the word cloud rendering and the visualization of the user snippets.

Conclusion and Future Work

This work presents a way to visualize similarities between new search results and already stored information. It also presents a possibility to access stored information without a context change for the user. The integration of the word cloud and the list of database snippets on the search engine results page offers a new possibility to interact with the information collage environment.

Although the feedback for the extension was good, it could not be proofed that the usage of the extension helped users to find more interesting search results and to speed up re-accessing their stored information. The perceived usefulness of the prototype may be too dependent on the user's individual browser database entries and a long term study has to be conducted to proof this hypothesis.

As a future development, further interaction possibilities with both visualization methods, the word cloud and the snippet list, should be considered. Also the shortcomings in the differential word cloud layout in the context of reproducing the word cloud with the same data set, could be targeted in future.

The selection of the relevant snippets from the user's locally stored data to the query terms strongly depends on the extracted keywords. Clearly, the retrieval of snippets from the local database has to be revised in the future. Also the keyword extraction and the weighting scheme for the selection of the most important keywords could be addressed for example by using a topic modelling.

List of Figures

1.1	Concept of berrypicking in a information retrieval process [Bat89].	1
1.2	Overview of activites of exploratory search [Mar06].	2
1.3	Model of sensemaking and foraging loop according to Pirolli and Card [PC05].	2
1.4	Growth rate of a typical user's bookmark archive [ABC98].	3
1.5	Visualization of search results.	4
3.1	Visualization of the data in the information collage prototype [Wal]. . . .	9
3.2	Working pipeline with user and computational steps.	10
4.1	Pipeline of processing the web search results.	13
4.2	Illustration of cosine similarity between documents [Pre08].	14
5.1	Linear Mapping of the corpus association to the x-position of a node. . .	16
5.2	Concept and implemenation of the word cloud layout.	17
5.3	Visualization of stored snippets	18
6.1	Concept of communication of content and background script [Zin15]. . . .	19
6.2	Code snippet of initializing the simulation and the collision detection. . .	20
7.1	Interpertation of SUS score [BKM09].	22

Bibliography

- [ABC98] David Abrams, Ron Baecker, and Mark Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 41–48, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
- [B⁺96] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [Bat89] Marcia J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989.
- [BGN08] Scott Bateman, Carl Gutwin, and Miguel Nacenta. Seeing things in the clouds: the effect of visual features on tag cloud selections. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 193–202. ACM, 2008.
- [BKM09] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.
- [Bos17] Mike Bostock. D3 data driven documents. <https://d3js.org/>, 2017. [Online; accessed March 2018].
- [BSA93] C. Buckley, G. Salton, and J. Allan. The smart information retrieval project. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 392–392, Stroudsburg, PA, USA, 1993. Association for Computational Linguistics.
- [CS14] Quim Castella and Charles Sutton. Word storms: Multiples of word clouds for visual comparison of documents. In *Proceedings of the 23rd international conference on World wide web*, pages 665–676. ACM, 2014.
- [DCCW08] Marian Dörk, Sheelagh Carpendale, Christopher Collins, and Carey Williamson. Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.

- [DES⁺15] Nicholas Diakopoulos, Dag Elgesem, Andrew Salway, Amy Zhang, and Knut Hofland. Compare clouds: Visualizing text corpora to compare media frames. In *Proceedings of IUI Workshop on Visual Text Analytics*, pages 193–202, 2015.
- [FFB18] Cristian Felix, Steven Franconeri, and Enrico Bertini. Taking word clouds apart: An empirical investigation of the design space for keyword summaries. *IEEE transactions on visualization and computer graphics*, 24(1):657–666, 2018.
- [fHaGMU06] Roy Rosenzweig Center for History and New Media at George Mason University. Zotero. <https://www.zotero.org/>, 2006. [Online; accessed March 2018].
- [JBR⁺18] Stefan Jänicke, Judith Blumenstein, Michaela Rücker, Dirk Zeckzer, and Gerik Scheuermann. Tagpies: Comparative visualization of textual data. In *VISIGRAPP (3: IVAPP)*, pages 40–51, 2018.
- [JML⁺18] Markus John, Eduard Marbach, Steffen Lohmann, Florian Heimerl, and Thomas Ertl. Multicloud: Interactive word cloud visualization for the analysis of multiple texts. In *Proceedings of Graphics Interface 2018*, GI 2018, pages 34 – 41. Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, 2018.
- [KHGW07] Byron YL Kuo, Thomas Hentrich, Benjamin M Good, and Mark D Wilkinson. Tag clouds for summarizing web search results. In *Proceedings of the 16th international conference on World Wide Web*, pages 1203–1204. ACM, 2007.
- [LHB⁺15] Steffen Lohmann, Florian Heimerl, Fabian Bopp, Michael Burch, and Thomas Ertl. Concentri cloud: Word cloud visualization for multiple text documents. In *Information Visualisation (iV), 2015 19th International Conference on*, pages 114–120. IEEE, 2015.
- [LZT09] Steffen Lohmann, Jürgen Ziegler, and Lena Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. In *IFIP Conference on Human-Computer Interaction*, pages 392–404. Springer, 2009.
- [Mar06] Gary Marchionini. Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41–46, April 2006.
- [Mee17] Elijah Meeks. d3-bboxcollide. <https://github.com/emeeks/d3-bboxCollide>, 2017. [Online; accessed March 2018].
- [Nie] Jacob Nielsen. Why you only need to test with 5 users. <https://www.nngroup.com/articles/>

- pwhy-you-only-need-to-test-with-5-users/. [Online; accessed July 2018].
- [NXB⁺16] Phong H. Nguyen, Kai Xu, Andy Bardill, Betul Salman, Kate Herd, and B. L. William Wong. Sensemap: Supporting browser-based online sensemaking through analytic provenance. In *2016 IEEE Conference on Visual Analytics Science and Technology, VAST 2016, Baltimore, MD, USA, October 23-28, 2016*, pages 91–100, 2016.
- [NXW⁺15] Phong H. Nguyen, Kai Xu, Ashley Wheat, B.L. William Wong, Simon Attfield, and Bob Fields. Sensepath: Understanding the sensemaking process through analytic provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22:41–50, 2015.
- [PC05] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of international conference on intelligence analysis*, volume 5, pages 2–4. McLean, VA, USA, 2005.
- [Por80] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [Pre08] Cambridge University Press. Dot products. <https://nlp.stanford.edu/IR-book/html/htmledition/dot-products-1.html>, 2008. [Online; accessed September 2018].
- [RGMM07] Anna W Rivadeneira, Daniel M Gruen, Michael J Muller, and David R Millen. Getting our head in the clouds: toward evaluation studies of tagclouds. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998. ACM, 2007.
- [Rot17] Tom Roth. Basics of d3 force directed graphs. <http://www.puzzlr.org/basics-of-d3-force-directed-graphs/>, 2017. [Online; accessed March 2018].
- [SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.
- [SCH08] James Sinclair and Michael Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15–29, 2008.
- [SM86] Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.
- [sol16] solariz. German stopwords list. https://github.com/solariz/german_stopwords, 2016. [Online; accessed March 2018].

- [Sto02] Sabine Stoessel. Methoden des testings im usability engineering. In *Usability*, pages 75–96. Springer, 2002.
- [SWY75] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [VWF09] Fernanda B Viegas, Martin Wattenberg, and Jonathan Feinberg. Participatory visualization with wordle. *IEEE transactions on visualization and computer graphics*, 15(6), 2009.
- [Wal] Manuela Waldner. Visual information foraging on the desktop. <https://www.cg.tuwien.ac.at/research/projects/deskollage/>. [Online; accessed March 2018].
- [Zin15] Aleks Zinevych. Communication chrome extension. <https://www.slideshare.net/OleksandrZinevych/chrome-extensions-56125231>, 2015. [Online; accessed September 2018].