Developing a Web Search Interface for the Visually Impaired

Author:
T.C. Beinema

Internal supervisors:
Dr. L.G. Vuurpijl
Dr. J.H.P. Kwisthout

External supervisor:
R.C.W. Versteeg, MSc
# Contents

1 Introduction ............................................................................................................. 5
   1.1 Visual Impairment .................................................................................................... 5
   1.2 Computer Usage ...................................................................................................... 5
   1.3 A Search Engine for the Blind ................................................................................ 6

2 Background ............................................................................................................. 9
   2.1 Screen Reader Browsing ........................................................................................... 9
   2.2 Usability and Accessibility ...................................................................................... 11
   2.3 Problems With Web Interactions ............................................................................ 12
      2.3.1 Problems With Websites in General .................................................................. 12
      2.3.2 Examples From the Literature - Websites in General ....................................... 14
      2.3.3 Problems With Online Search ........................................................................... 15
      2.3.4 Examples From the Literature - Online Search ................................................ 17
   2.4 Browsing Strategies and Coping Behaviour .......................................................... 18
   2.5 Guidelines for Web Development .......................................................................... 19
      2.5.1 Guidelines for Websites in General .................................................................. 19
      2.5.2 Guidelines For Online Search .......................................................................... 20
      2.5.3 Applicable Guidelines ...................................................................................... 21
   2.6 Search Engine Implementations ............................................................................ 24
   2.7 Summary ................................................................................................................. 26

3 Design ................................................................................................................... 27
   3.1 Scope ....................................................................................................................... 27
      3.1.1 Target Group ..................................................................................................... 27
      3.1.2 Application Type .............................................................................................. 27
      3.1.3 Search Results .................................................................................................. 28
   3.2 Minimal Application Features .............................................................................. 28
   3.3 Requirements .......................................................................................................... 28
   3.4 Content Design ...................................................................................................... 29
      3.4.1 Page Content Design ....................................................................................... 29
      3.4.2 Search Results .................................................................................................. 30
      3.4.3 Clustering ......................................................................................................... 31

4 Clustering ............................................................................................................... 32
   4.1 Algorithm Requirements ....................................................................................... 32
   4.2 Clustering Algorithms ............................................................................................. 33
   4.3 Lingo - How Does It Work? .................................................................................. 33
      4.3.1 Step 1: Pre-processing ........................................................................................ 34
      4.3.2 Step 2: Phrase Extraction .................................................................................. 35
      4.3.3 Step 3: Cluster Label Induction ......................................................................... 36
      4.3.4 Step 4: Cluster-Content Allocation .................................................................... 39
      4.3.5 Step 5: Final Cluster Formation ........................................................................ 40
   4.4 Resorting of Clusters .............................................................................................. 40
Preface

Before you lies the thesis “Developing a Web Search Interface for the Visually Impaired”. This thesis is based on the research and development of a proof of concept application that could provide the blind with the same information as the sighted in a more accessible manner. It was written to fulfill the graduation requirements of the Web and Language Interaction track of the Artificial Intelligence Masters Programme at the Radboud University, Nijmegen.

The research project described was my assignment during a six-month internship with Dedicon, a Dutch foundation that creates and provides tools and materials for people with reading disabilities. Examples of these materials are braille books, 3D-maps, and karaoke books (which are a combination of audio books and ebooks that enables a reader to listen to the text that they are reading while the corresponding words light up simultaneously). The research project was challenging since there were many directions in which the research could proceed. Fortunately my supervisors Arjan, Reinout, Louis and Johan could always provide me with good advice and were able to help me find the correct direction in which to proceed. I would like to thank them for their guidance.

I would also like to thank my colleagues at Dedicon for the great working environment. Our daily walks in the break and conversations made my internship a great experience. I would also like to thank Davy for his insights and advice on the application and for the time he was willing to spend. I would also like to thank Dedicon’s survey expert for the advice on the development of the survey and their external communications people for helping me to reach the survey’s respondents. Also, many thanks to all the respondents to my questionnaire. Finally, I would like to thank my parents, boyfriend and siblings for their patience, encouragement and willingness to proofread various parts of my thesis.

To you as a reader: Enjoy your reading!

Tessa Beinema

Nijmegen, March 30, 2017
Abstract

In this thesis the development process of a proof of concept for a web search interface for the blind is described. The need for such an application was confirmed in a survey that was conducted. It was found that blind and sighted users both performed an equal number of searches, but that the blind were often less successful at finding the information they were looking for. The development process consisted of several parts. Based on an extensive review of the literature and requirements, guidelines were formulated. This was followed by the design and implementation of an interface, that has a structure optimized for use with a screen reader. The interface shows clusters with search results that are returned for a user’s query. The evaluation of the interface was done through expert reviews. Three experts tested the application using a screen reader and operating system of their own choice. Their findings were overall positive, that is, the interface confirms to the formulated guidelines, is intuitive and shows promise for improving the search process for the blind. This thesis concludes with suggestions for future research.
1 Introduction

Using a computer is part of our daily lives. Ever since the 1970s, the control of these machines has become more and more graphical. For the average user this has been an improvement since textual interfaces provide the user with fewer visual cues. There is however also a fair percentage of users for whom a graphical user interface makes using a computer even more of a challenge. This is true, for example, for the visually impaired.

1.1 Visual Impairment

Visual impairment is defined as a loss of vision that is not correctable - for example, by wearing a pair of glasses or lenses. Kleynhans and Fourie (2014) stated that the terms ‘visually impaired’, ‘partially sighted’ and ‘low vision’ sometimes seem to be used interchangeably. They then proceed by clarifying how ‘visually impaired’ and ‘blind’ can be distinguished starting with a definition by the World Health Organisation (WHO). This definition states that ‘visual impairment’ is used to indicate the category of impairment in both low vision and blindness. Low vision indicates an impairment in the range between 3/60 and 6/18, where the first number indicates the distance in feet at which a person can see what a non-impaired person can see at the distance in feet of the second number. People who are classified as blind can see less than 3/60. The international Statistical Classification of Diseases divides visual impairment in terms that range from ‘mild visual impairment in one eye’ to ‘total blindness’ (Dandona & Dandona, 2006). The WHO fact sheet on visual impairment states that the number of visually impaired people in the world is estimated at 285 million - that is, 39 million blind and 246 million with low vision (World Health Organization, 2014). Also, over 82% of the blind are over 50 years old. In Europe this is estimated at 30 million visually impaired, of which 2.5 million are estimated to be blind (European Blindness Union, 2015). In the Netherlands there are currently 350,000 visually impaired and their number is estimated to grow to 380,000 in 2020 (Bartiméus Sonneheerdt, 2016). In this thesis three groups will be distinguished, namely: the blind (‘no vision’), the partially sighted (‘poor vision’), and the sighted.

1.2 Computer Usage

When using a computer, there are a number of reading aids that a visually impaired user can employ. If a user still has some sight left they can use a magnifying glass or special contrast settings (Rubin & Legge, 1989), but if this does not provide them with enough help - for example, when their vision is too limited or using the named options is very tiring - a user might also use a screen reader. A screen reader can either read out the information displayed on the screen or output it to a refreshable braille display.

While blind users have incredible skills in using their screen readers, and are able to solve many problems they encounter, it can still be very difficult for them to use a computer. The main problem is the large variability between various computer programs. Even though there is a general scheme that programs use, they vary in terms of, for example, content and menu items. Even without the content or unexpected pop-ups it can be hard to figure out for a visually impaired user where they are in an application and what is happening.

Something that can be even more difficult than using local programs is browsing the internet. Since there is no standard template for websites, there is a huge variety in website layouts. This can make it difficult for the visually impaired to find the information they are looking for on the web. Even if a search engine is used it can be difficult to find the link to a website that might contain the sought for information.
The large variability on the internet can put a visually impaired user at a disadvantage since the internet has become one of the most used sources of information. While a user who can use magnifying software or enhanced contrast may still use some of the visual cues that are available on a website, a user who uses a screen reader has only the vocalised information to navigate on. Browsing the web requires some navigational skills, and even when users have acquired these, certain problems are hard to overcome, especially when there are many different types of websites and layouts. Harper, Goble, and Stevens (2001) provided their readers with an example that allowed them to experience (some of) the problems that a visually impaired user might encounter when viewing a web page using a reading aid such as a magnifying glass. The instruction for this example was for the user to ‘[...] start their browser and limit the window size to the top left fifth of the screen.’ and to then try and perform some standard online interactions. When imagining a blind user, such an example for using a screen reader might be to open the web page’s HTML code and try and quickly find the information that you are looking for. As can be imagined, this has a higher difficulty to it than searching information on a website with all of the cues available.

![Diagram](https://via.placeholder.com/150)

*Figure 1: A scheme that depicts the various parts of the web search interface on which research might be performed and where evaluation might take place. The labels and connections represented in this scheme will be discussed in the main text.*

### 1.3 A Search Engine for the Blind

In view of the increasing number of people who are visually impaired, the importance of the internet, and the difficulties that this user group experience while searching for information, having a search engine geared to their needs seems an exigency. A survey that we conducted among the blind in the Netherlands confirmed this (for a detailed description of its construction, questions, results, and conclusions see Appendices A and B). It was found that blind respondents performed as many
online searches per day as sighted respondents, but that they reported that they were less likely
to find what they were looking for in one attempt. This indicates that there is a gap between
the sighted and the blind, when it comes to the ease of finding information online. It can be
hypothesised that this the cause for this difference might lay in the interaction with the search
interface that the participants used, since the main difference between the groups is the use or lack
of use of a screen-reader.

In the research project, performed at Dedicon (a Dutch foundation that creates and provides
tools and materials for people with reading disabilities), which led to this thesis it was investigated
whether a web application can be built that might provide the blind with the same information
as the sighted in a more accessible manner and what would be a suitable interface for such an
application. When observing a search interface, there are several parts of that interface on which
research can be conducted. To illustrate this, a schematic overview of the various parts that make
up a web search interface can be found in Figure 1. As can be seen, a user is depicted (labelled ‘1’)
who has an information need and submits a search question (query) to the web search interface
(labelled ‘2’). The query is then put through to the state-of-the-art existing search engine from
which results are retrieved. Improvements to the generation process for search results by the state-
of-the-art search engine itself are left outside of the scope. We receive a set of results for the
information need and these are then displayed (either at ‘3a’ or ‘3b’). In the case of ‘3a’ we would
focus on presenting the clustered results in a most optimal manner. In the case of ‘3b’ features
would be added that would help the user to specify their query based on the presented results. An
extra feature at the user side of the process (for example, when they submit their query) could be to
provide them with extra tools to help them specify their query. There are three evaluation options
that correspond to the various research possibilities (labelled ‘A’, ‘B’, and ‘C’). In ‘A’ evaluation
could be performed by comparing the effectiveness of the queries of a group of users that use the
extra tools for query formulation to those of those that do not use these tools. The evaluation in
‘B’ and ‘C’ could be to have a group of experts review the usability of the web search application
or to perform a large scale user study.

Figure 2: A schematic representation of the steps performed in this research project and their
results.

In this project we have decided to focus on the accessibility and usability of the interface, since
we believe that much can be won by creating a more optimal interaction with the search interface
and an improved presentation of the results for users of a screen reader. To develop a proof of
concept several steps were taken (see also Figure 2). First, the current manner of searching on the
web by the blind and common problems in that process were extensively researched in a literature
study to acquire insight into the target group’s experiences and skills. This literature study also
resulted in a set of requirements and guidelines that should be followed to improve usability for the
blind. In the second step, additional requirements were formulated, many considerations were made
and these were incorporated in a design. Since it was concluded from the literature study that one
of the key factors for usability would be to have a clear structure and that this could be achieved by showing the search results as clusters, the third step consisted of a second literature study, which was performed to find a suitable clustering algorithm for that task. During the literature research for a suitable algorithm, requirements and preferences for the algorithm were defined and a table was constructed in which algorithms were compared. In the fourth step, after the algorithm had been found, the application design was implemented while consulting with a blind expert and its parameters were tuned based on the survey held amongst the target group. At step five, the final version of the proof of concept application was evaluated by experts. Finally, conclusions were formulated and suggestions were made for future research.

In the following chapters the process for each of these steps will be described. First an extensive review of the literature will be given to provide more insight in the experiences of the blind on the web and on already available development guidelines (Chapter 2). This review will be followed by information on the design of the application based on the requirements and guidelines from the literature (Chapter 3). In Chapter 4 a short overview of possible clustering algorithms will be given and it will be explained how the chosen algorithm works. Following that chapter the implementation and evaluation process of the web application are described (Chapter 5 and 6, respectively). Finally, the conclusions that can be taken from the development process and evaluation results will be stated and discussed, and recommendations for future work will be presented (Chapter 7).
2 Background

The blind apply a number of reading aids to enable them to use a computer. The most used reading aids are screen readers, which can vocalize the information on the screen, but can also output the information to a refreshable braille display if it is available. Still, when using a computer, blind users run into problems. In the following subsections a more detailed description of the use of screen readers and the problems encountered will be given based on the literature. This will be followed by information on how these problems are dealt with by blind users and how they might be prevented when building websites. This chapter concludes with a summary of the problems with their possible solutions. These solutions (for example, following certain guidelines) can be seen as requirements that could be used when designing a website (or in our case: search engine) for the blind.

2.1 Screen Reader Browsing

There are several screen readers that a blind user can chose from. Both Windows and OS X come with a pre-installed screen reader (Microsoft’s Narrator (Microsoft, 2015) and Apple’s VoiceOver (Apple, 2015) respectively). For Windows, there are also commercial options of which the most well-known are JAWS (Freedom Scientific, 2015), SuperNova (Dolphin Computer Access Ltd., 2015), NVDA (NV Access, 2015), and Window-Eyes (GW Micro Inc., 2015). From these four only NVDA is freely available.

```html
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="mystyle.css">
  </head>
  <body>
    <ul class="menu" id="mymenu">
      <li><a href="#home">Home</a></li>
      <li><a href="#news">News</a></li>
      <li><a href="#contact">Contact</a></li>
      <li><a href="#about">About</a></li>
    </ul>
    <img src="lens.jpg"></img>
    <h1>The Lorem Ipsum Text</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. </p>
    <p>Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo.</p>
  </body>
</html>
```

Figure 3: Example HTML code, which corresponds to the web page in Figure 4.
Andronico and Buzzi (2005) describe how a screen reader processes a web page (in their case JAWS). This basic navigation is similar in all of the screen readers (Borodin, Bigham, Dausch, & Ramakrishnan, 2010). When processing a page, a screen reader will start at the beginning of the HTML code that makes up the web page and will then work its way towards the end. Users can navigate the web page using the arrow keys, the Tab key or special JAWS commands - in a similar manner as in which someone would move through a text file without using a mouse. One of the major differences between blind and sighted users is that the visually impaired usually do not use the mouse (Leporini & Paternò, 2008). Also, the item that the user is looking for might be included at the end of the HTML text. Such elements that seem well-placed and are easy to locate for a sighted user, due to the layout added using CSS, can very well be hard to find for a visually impaired user (Leporini, Andronico, & Buzzi, 2004). In Figures 3 and 4 the HTML source (Figure 3) and the web page as displayed to the user (Figure 4) are shown for an example web page.

As can be seen, the interpretation of the HTML code and the addition of CSS provides a sighted user with a lot of visual cues, which help them to build a mental model of the web page faster. If, for example, a sighted user would be asked to look at the second paragraph of text they would be able to directly look at it since the main text can be easily located and there is a small space between the two paragraphs. A blind user, on the other hand, would first have to navigate past the menu items, the image and the header text before getting to the paragraphs.

Some screen readers also have advanced features. JAWS, for example, has features that make use of certain HTML tags, which can be very useful if these tags are present on a page (Andronico & Buzzi, 2005). An example of such a tag are heading tags (ranging from \(<h1>\) up and to \(<h6>\)) by which titles of sections/subsections are labelled. These tags can be used by JAWS to make an index of the web page, which allows users to find the section of their interest faster. In, for example, NVDA a user can skip through all the elements of a certain type (buttons, links, headings, etcetera) (Borodin et al., 2010). There are also screen readers that provide the option to navigate through these interactive elements using a screen reader shortcut and a specific letter for each type of element.

A last feature found with some of the screen readers is that they have two or more edit modes,
which allows the user to switch between navigating through elements and interacting with an
element (for example, skipping to the next text box or typing text in a text box) (Borodin et al.,
2010).

Having a screen reader with special features, such as mentioned above, provides the visually
impaired user with a great asset, but the effectiveness of such an aid is still very dependent on
the design of the website that they are accessing, as we saw in the example web page and HTML
code. For example, in the example page the title is neatly labelled with <h1>-tags, but in a poorly
designed website a title could also be created by taking a part of the text and enlarging its font
size in the CSS file. The title would then no longer have the tags that mark it as a title, which are
used by the screen reader’s (special) features.

2.2 Usability and Accessibility

The usability of a website with a screen reader depends on a number of factors that are mostly
related to the website’s accessibility. However, having an accessible website does not necessarily
mean that it is also usable. In the literature, guidelines can be found that provide cues as to how
to develop accessible websites. Notable examples are Nielsen (1994), Chisholm, Vanderheiden, and
Jacobs (2001), and Caldwell, Cooper, Reid, and Vanderheiden (2008)\footnote{Since there are many papers that provide guidelines and discuss problems more references are provided in the overview sections of problems and guidelines, which are Sections 2.3 and 2.5 respectively.}. The papers in which these
guidelines are proposed also give examples of problems that users can have with websites. In the
following Section (2.3) an overview of these problems will be given. Section 2.5 elaborates on the
various guidelines.

An important notion when talking about the usability and accessibility of websites is the dif-
ference in meaning between these terms. As was noted in a large number of papers: an accessible
website might also be usable, but a usable website must be accessible. That is, having a website
that conforms to all the accessibility standards does not necessarily mean that the usability of that
website is also guaranteed. As Theofanos and Redish (2003) state ‘[…] if a website is not usable,
it is not really accessible […]’. This difference that can exist between having accessibility and
reaching actual usability is also known as the ‘gap between accessibility and usability’. Leuthold,
Bargas-Avila, and Opwis (2008) mention that among researchers there has been a tendency to focus
on making information and functionality accessible for visually impaired users, while forgetting to
follow so called user interface guidelines - or at least: they did not apply them. The gap between
usability and accessibility can be illustrated using two examples. The first is the following: You
can have a website that conforms to all the accessibility guidelines, but this does not mean that
a user can navigate the site well and actually find what they are looking for. As a result visually
impaired users are being limited in the amount of information they can access, since they cannot or
will not use the websites with accessibility problems (Borodin et al., 2010). The second example is
the following: A site has added alternative descriptions for images (so called: ALT-text), but there
are too many images and the descriptions are too long or irrelevant for the content (‘decorative
image 5’), which makes the actual information on the site difficult to process for a visually impaired
user. Something that can also happen is that at first, websites might conform to the guidelines, but
no longer conform to them when they are developed further later on (Lazar, Allen, Kleinman, &
Malarkey, 2007). Even when the conformance of a website was tested automatically on accessibility
it can still prove to have poor usability (Vigo, Brown, & Conway, 2013).

To improve usability several researchers have suggested in their papers that the development of
websites should not only include conformance to guidelines, but that the focus should also be on
making websites accessible from the start of their development (Power, Freire, Petrie, & Swallow,
2012; Rømen & Svanæs, 2012; Kelly, Sloan, Phipps, Petrie, & Hamilton, 2005). Rømen and Svanæs (2012) state that a user-centered design approach and usability tests with specified users might provide insights as to how to improve the accessibility of websites.

2.3 Problems With Web Interactions

As mentioned previously, browsing the web can be difficult for a blind user. It is therefore recommended to perform extensive background research so that an idea can be formed as to which problems might be expected to occur that could easily be avoided. There are quite a number of papers that describe the problems that blind users encounter when browsing in general. There are also some papers that specifically focus on the use of search engines by the blind. Below, we will first give an overview of problems that occur when browsing the web in general. Next, we provide an overview of the problems that occur more specifically when searching on the web.

2.3.1 Problems With Websites in General

Even though the use of various computer aids already in itself makes websites more accessible, there are still many parts of the web browsing process that are complicated. In the literature there is a general consensus on what constitutes the main problems that visually impaired users - and more specifically the users of screen readers - encounter. The problems fall into four groups:

- Lack of context
- Sequential navigation
- Cognitive load
- Screen reader interpretation

Each of these will be described in the following subsections.

Lack of Context

A visually impaired user only has access to a small portion of the web page at a time. This can cause them to lose track of the overall layout and context of the page (Andronico & Buzzi, 2005; Buzzi, Andronico, & Leporini, 2004; Correani, Leporini, & Paterno, 2004; Leporini et al., 2004; Leporini & Paternô, 2008). Or, as Harper et al. (2001) describes, users cannot get a feel for what is on the page, they do not know where they are and they get disoriented. To give an example, elaborating on the example given in the introduction: Imagine that you have to find some information on a web page, but that you only have access to the HTML representation of that web page. Also, imagine that you are not able to read the HTML yourself, but that someone is reading it to you. They might have started at the top of web page, but they might also have started somewhere in the middle because there was something on the web page that told them to do so. This could, for example, occur if a website has a form that is designed to automatically focus on the first text input field. Another, more figurative, example could be the one displayed in Figure 5. In both examples, there is not only a lack of context (‘Where on the page am I?’), but a screen reader will simply read the text to you in a linear manner, which introduces yet another problem, known as sequential navigation.

Sequential Navigation

Users of screen readers mostly navigate by keyboard and not by mouse (Andronico & Buzzi, 2005; Buzzi et al., 2004). This inability to use a mouse is also called ‘the mouse barrier’ and is a psychomotorical problem caused by a lack of feedback on the movements that are performed (Donker,
The presentation of the content on a web page that blind users experience requires a high-cognitive effort from screen reader users (Gooda Sahib, Tombros, & Stockman, 2012). They are presented with layout information, as well as (ir)relevant content in a linear manner. A blind user has to build a mental representation of the web page. For sighted users the page has many visual cues that inform them about the role and grouping of items (for example, menu items), but for blind users this information is not available (Leuthold et al., 2008). The linear information, and having to remember what the location is of the screen reader’s cursor on the page and which elements they have already encountered, makes that a lot has to be remembered. Blind users also have to predict which elements they might encounter or where they might find the thing they are looking for (Andronico & Buzzi, 2005). A user also has to know the commands for the use of the screen reader (on top of those for the use of a computer), which requires considerable effort (Borodin et al., 2010). Therefore, most blind users tend to employ basic commands, and web navigation is often considered to be a time-consuming, boring and stressful experience (Leporini, Andronico, Buzzi, & Castillo, 2008).

Oviatt (2006) describes various human-centered design principles for developing applications that aim to minimize the cognitive load that users experience. The goal of human-centered design is to design interfaces that are intuitive, easy to learn and free of errors. In order to do so, the behaviour of the user and responses that can be expected have to be analysed beforehand. Oviatt names eight design principles that can be followed in the development process. These include, for example, ‘Accommodate user’s existing familiar work practice rather than attempting to change it.’ and ‘Minimize cognitive load associated with extraneous complexity of system output.’. In the case of blind users their ‘existing familiar work practice’ includes the use of a screen reader, which
brings us to the final groups of problems in our list.

**Screen Reader Interpretation**

Reading a web page using a screen reader means processing a lot of intertwined types of information. This can be caused through various types of structures or element usage. An example for this can be pages that contain many elements that are repeated on each page of a website. Users could find themselves in a situation in which they have to listen to all the menu items, advertisements, and layout images before reaching the information that the specific page is actually about (Andronico & Buzzi, 2005; Buzzi et al., 2004; Correani et al., 2004; King, Evans, & Blenkhorn, 2003; Leporini et al., 2004; Leporini & Paternò, 2008; Murphy, Kuber, McAllister, Strain, & Yu, 2008). This can lead a user to experience that there is too much detail for what they can observe and that the web page is too complex (Harper et al., 2001). Sometimes web pages have so-called ‘skip links’, which are links incorporated in the web page that are not visible for a sighted user, but which are read out by a screen reader. These links allow the user to skip, for example, a menu and go directly to the main content of the page (Theofanos & Redish, 2003).

Another example could be a web page such as Wikipedia that contains a lot of links in its text. When narrating that type of web page a screen reader would start with the text, but it would interrupt the flow every time it encountered a link to announce that a link was found. A user of Apple’s VoiceOver would, for example, hear ‘A - link - screen reader is [...]’ for a web page that has a sentence ‘A screen reader is’ in which the words ‘screen reader’ link somewhere. If an entire web page is read out in that manner, the user has to put in a lot of effort to follow what the complete sentences are that are being read (Borodin et al., 2010). In the paper by Vigo et al. (2013) users are being quoted saying “This is a problem sometimes, you can have 30 or 230 links that you have to sit and listen to!”, and “900 links ... OK that’s enough.”.

A third example can be the occurrence of dynamic content such as banners or pop-up boxes that reset the focus of the screen reader back to the top of the page (King et al., 2003; Lazar et al., 2007).

Finally, the absence of alternative text for, for example, images (Murphy et al., 2008) and the presence of elements that are unidentifiable from how they are read out can also be problematic (Lazar et al., 2007).

For all of the types of problems with screen readers described above the literature presents ample real life examples. Below we first give some examples of problems with websites in general. Next we focus more specifically on what the literature has to say about online search.

### 2.3.2 Examples From the Literature - Websites in General

Many papers provide examples of blind users experiencing problems on the web. These range from the more general problems with websites that were mentioned above to problems with specific parts of websites, such as forms and CAPTCHAs. These descriptions give a good insight into what the difficulties users of screen readers experience in reading web pages.

Babu, Singh, and Ganesh (2010) asked six blind students to try and use a so called ‘test manager’ that was embedded in the Blackboard LMS. The students were asked to fill in a quiz using this test manager and while doing so had to vocalize their actions and thoughts. A problem encountered in the analysis of the recorded speech (both from participants and their screen reader) included the accidentally skipping of a question. One of the main causes for this was that, when a participant would go to the next question, the text of button for doing so would also be read again when on the next page. The user would then think that the attempt had failed and try again, thus skipping a question. Another difficulty that users faced was when they were expected to answer an open
question instead of a multiple choice question - since it was not indicated what type of question was being asked. There was also a case in which the user accidentally stopped taking the quiz because he pressed a button that did not go well with the interface. He was unable to take the quiz again because you were only allowed to take it once.

Visually impaired users also often have difficulties with the security measures that are used on the web. Dosono, Hayes, and Wang (2015) researched the problems that were encountered in authentication by people with a visual impairment. They observed and interacted with 12 participants performing a series of web tasks. In their results section they give examples of problems encountered. For example, when a user was looking for a ‘login field’ by looking for links starting with ‘l’ she could not find the login field because it was called ‘sign in’. There is also an example of a participant who gets frustrated because the website had been changed and she could not find the correct link in her search results. A second thing that happened to that participant was that she tried to login to her account at the site of her bank, but was actually trying to use the fields for the Google services. Frustrations were also expressed about JAWS. These focussed on the lack of output for error messages and the masking of passwords, and lack of feedback for entering case-sensitive passwords. Fuglerud (2011) found that their Norwegian participants would typically let a family member do their online banking for them.

A specific example of an authentication technique that is impossible to use for visually impaired users are CAPTCHAs, which are pictures from which a series of characters has to be typed into a box. Studies by Murphy et al. (2008) and Fuglerud (2011) mention that it was very difficult for visually impaired users to register for, for example, email accounts or login to banking accounts because they have to fill in a CAPTCHA. Bigham and Cavender (2009) researched the usability of audio CAPTCHAs, which were designed to aid the visually impaired (Schlaikjer, 2007), both for sighted and visually impaired users. In an audio CAPTCHA there is a spoken word with noise added that has to be typed into a box. They performed a study with 162 participants and found that audio CAPTCHAs were substantially more difficult than normal CAPTCHAs for both groups. Only 39% and 43% of the audio CAPTCHAs were solved at the first try for sighted and blind users, respectively, but blind participants did take twice as long to do so. Most of the unsolved CAPTCHAs were not solved at the second or third try either and thus never solved. Visually impaired users mainly had difficulties with hearing the CAPTCHA and their screen reader at the same time, and with having to turn on the CAPTCHA and then quickly going to the field where they had to fill in the answer.

2.3.3 Problems With Online Search

Exploration of the web is difficult for a blind user (Andronico & Buzzi, 2005; Gooda Sahib et al., 2012; Gooda Sahib, Stockman, Tombros, & Metatla, 2013). As stated in the introduction, a type of website that is especially difficult to use for a blind user are search engines. Bigham and Cavender (2007) found that while it took sighted users 34.81 seconds on average to choose a first result to look at while for blind users it took 155.06 seconds. There also was a significant difference in submitting a query - 34.54 seconds and 74.66 seconds, respectively. Compared to sighted users the blind have to perform extra tasks to use a search engine (Brajnik, 2004). These are tasks that are needed to navigate using the screen reader, submit the query and especially in exploring the results. These search engine specific problems are added to the general web problems mentioned above. Examples of specific features that make searching difficult are, for example:

- Dynamic suggestions.
- Advertisements
- Results ordering
• ‘Unreachable’ features

The problems with each of these will be clarified in the following subsections.

**Dynamic Suggestions**

Dynamic suggestions are suggestions that are being made by the system while you type in your search question (query) (see Figure 6). For a sighted user, a list is shown that updates with every character that you add to or remove from your query. If blind users were to use this feature, they would have to go into that list, observe, and go back to the search field where they could add a new letter. They could repeat this process for every character that they add. A manner to cope with this could be to just ‘not use’ these suggestions. However the drop down list with suggestions also interferes when they were to try and use their arrow keys to move the focus from the search field to the ‘search’ button. This might result in changing their query unintentionally.

**Advertisements**

Advertisements are included in the search results lists of, for example, the most popular search engines Bing and Google. Google places an ‘ADV’ marker in front of the link and places advertisements at the top of the results list, but when skipping through the links in the results list the marker is not read out with the link title. Moreover, there is no indicator for when the advertisement part of the list is finished and the ‘normal’ links in the results list are read out. The screen reader makes them sound all the same.

**Results Ordering**

Apart from the advertisement links sounding the same as the ‘normal’ results in the results list, the list of results also provides the blind user with yet another ‘lack of context’ issue. A user has to find the search results that suggest they link to a web page that might provide the user with what he is looking for. However, a search term can return results that might actually be dividable into several sub-categories. This applies, for example, to words that are homonyms and/or homographs. The homonym ‘date’, for example, can result into results about the fruit (type 1), a time in the year (e.g. the 1st of July 2016) (type 2) or a romantic meeting (type 3). A results list could then have the following order: 1, 1, 2, 3, 1, 2, 1, 3, 3, 2, etc. A sighted user looking for information about the fruit who would want to visit multiple web pages about it would probably perform keyword spotting, which enables them to quickly ‘jump across the page’. A blind user would have to listen to each result, decide if it is relevant, and continue to the next result. This means that to get to the third possibly relevant result they would have to listen to five links. This can be a highly ineffective process even if the links about the fruit are all relevant enough. It might be the case that one link talks about buying the fruit online, the other might be the Wikipedia description and the third might be an article describing how healthy they are to eat.

**Unreachable Features**

The last feature of search engines that make their use difficult for blind users are features that are ‘unreachable’. By this, features are meant that are there and are accessible, but only if a user knows where to find them. One such feature is the Advanced Search, which allows users to provide extra filters for the search results, such as a time frame or the size of an image (in case they are searching for images). The advanced search is easily accessible for sighted users since it is a button next to the tabs ‘All’, ‘Images’, ‘Videos’, and ‘Shopping’, but a blind user would be likely to skip to the contents of the results list either immediately or after encountering the categories ‘Images’ and ‘Videos’ and they will therefore never reach this option.
In the following section, we present some examples of these types of problems, that have been described in the literature, are given.

### 2.3.4 Examples From the Literature - Online Search

Many of the papers that describe research on making the web accessible/usable for visually impaired users provide examples of situations in which the interaction is unsatisfactory. In the following paragraphs an overview is given of some of the descriptions relating to search engines as can be found in the literature.

A first example is the hypothetical case of ‘Roberto’, which is used by Andronico and Buzzi (2005) to illustrate the problems that blind users might face when using a search interface. It is described how Roberto goes to the Google URL and starts searching for the so called ‘search box’. He uses the screen reader to skip through the links that are available on the page and finds something called an ‘edit field’, but he is unsure if this is the search box or another edit field. He then finds a ‘Google Search’-button next to it and can then infer that it was indeed the search box. The search results in the case of this particular version of the Google interface just consisted of a list of links. This made that the user had no means of skipping past all the non-useful links (for example, the Google account tools button where you can find a link to your Gmail/Drive/etcetera), and even if the user had passed all the irrelevant links the results were just that: a long list of links that had to be explored one by one.

Vigo et al. (2013) report about two studies in one which blind users encountered confusing situations. There was, for example, a blind user that searched for information, but no results were found.

Gooda Sahib et al. (2012) found that the query suggestions, spelling suggestions and related searches were almost unused by their blind participants even though about 60% of their subjects was aware of these features. Gooda Sahib et al. (2013) quoted a user stating "We hardly notice which term is misspelt." Gooda Sahib, Tombros, and Stockman (2015) stated that visually impaired users did not benefit from the real-time query suggestions that appear as the user types the query and therefore discover misspelled terms only when the query has been submitted. They also state that users do not ‘get’ spelling suggestions if the suggestion sounds the same as what they tried to type.
2.4 Browsing Strategies and Coping Behaviour

As discussed and illustrated in the previous sections, the usability of websites can be poor for the blind. In the literature it is shown that this can result in certain browsing behaviours and strategies, which can sometimes even be identified as coping strategies. Coping is defined in Vigo et al. (2013) as ‘Constantly changing cognitive and behavioural efforts to manage specific external and/or internal demands that are appraised as taxing or exceeding the resources of a person’ (a definition credited to: Lazarus (1993). As Lazar et al. (2007) state as a conclusion in their paper, blind users often tend to find an alternative solution to a problem more often than that they already knew how to solve a problem. They also found that blind users tended to be less likely to give up (that is, to restart a program or the computer) than sighted users. Blind users were also not quick to report that they felt angry at themselves, but more likely to report that they were determined to fix the problem.

For example, Bigham and Cavender (2007) describe several possible coping strategies that blind users apply when faced with inaccessible content (that is, Javascript, XML, Flash content, images without ALT text, etcetera). An important technique is so-called ‘probing’ (also named by Jobst (2009, p. 79) and Harper et al. (2001)). Probing can be described as following a link to see where it leads and then returning to where you were. Users apply this technique to see if the content in a link might be relevant, but they return when it proves not to be. This can happen, for example, if there is little context around the link to suggest the usability of its content (‘click here’, for example, is not that informative). In the research by Bigham and Cavender (2007) they found that their blind participants performed 0.34 probes per web page while the sighted participants performed 0.12, which was a significant difference. A user in the research by Vigo and Harper (2013) stated “I can’t read this but I will give it a try, I am assuming it is what I want so let’s see where it takes me.”.

Some users also change their location as a way of coping with confusion about their location on the web page. In the paper by Vigo and Harper (2013) users were said to either return, escape or backtrack. A user returning to the beginning of a web page when they got lost was quoted as saying: “Not sure where I am ... if in doubt go back to the beginning.” after all the results for their search query had been read. Escaping would be done by so-called ‘impulsive clicking’, that is, clicking any link to get away from the situation that they were in. The third option, backtracking, was applied by returning to a page that they had previously encountered when they had, for example, clicked a link that lead to a dead end page.

Borodin et al. (2010) provides a survey of screen reader browsing strategies. They describe that blind users increase the speech rate of their screen reader. For some experts the speech rate could be as high as 500 words per minute. Another type of strategy that is described is for users to apply a self-learned manner of getting an overview of the structure of the page and using that as a mental model of the website used for further exploration. This is for example done by first jumping from heading to heading to get an overview of which sections are present on a page. Users also use a similar tactic when, for example, having to fill in forms by pressing the Tab-key to jump from field to field. The jumping from heading to heading or from link to link strategy was also reported by Gooda Sahib et al. (2012). As a last resort, users might call on another person to do something for them (for example to help them to get past the earlier mentioned CAPTCHAs) (Borodin et al., 2010).

Dosono et al. (2015) name some alternative methods that they observed when visually impaired users had to authenticate themselves. These include the use of a password manager (for example, let the Google Chrome browser remember them and fill them in), and the use of biometrics (a fingerprint scanner) to unlock their laptop.
Finally, Gooda Sahib et al. (2012) found that visually impaired users used longer queries when searching. Visually impaired users used 4.61 (sd = 2.76) words per query while this number was 3.86 (sd = 0.67) for sighted users. The reason for the longer average formulation length by the visually impaired could be ascribed to the fact that more specific queries would result in the intended search result (that is, a web page that contains the sought information) being at the start of the results list.

2.5 Guidelines for Web Development

As mentioned in ‘Usability and Accessibility’ (Section 2.2), a lot has been written on accessible and usable websites. This has resulted in many sets of guidelines. There are guidelines that are for websites in general, and even specific guidelines for accessible search engines. These guidelines will be discussed in the following subsections.

2.5.1 Guidelines for Websites in General

There are several sets of guidelines for the development of websites. These range from the well known ‘10 heuristics of accessibility’ by Nielsen (Preece, Rogers, & Sharp, 2002, p. 27), which are very general, to the Web Content Accessibility Guidelines (WCAG), which are specifically aimed at making the web accessible for people with disabilities. Another method that could improve the accessibility of websites is to conform to the guidelines for the use of what is called WAI-ARIA markup. There are also many papers that make suggestions for accessibility guidelines. In the following three subsections the named guidelines will be discussed further starting with the WCAG and the WAI-ARIA markup since many of the guidelines from the papers are suggestions to be added to use of the WCAG.

**Web Content Accessibility Guidelines (WCAG)**

The Web Content Accessibility Guidelines (WCAG) are a set of guidelines put together by the World Wide Web Consortium (also known as the W3C). The first version of these guidelines dates from May 5th, 1999 (Chisholm et al., 2001). The guidelines were intended to make the web more accessible for people with disabilities. Since people with disabilities might not be able to use a keyboard or mouse, or they might use certain aids to view or listen to a website, guidelines that can help optimise a website for this type of user seemed useful. The key aim in the WCAG 1.0 was to help developers design web pages that could ‘transform gracefully’, which meant that they would stay accessible even if they, for example, contained a lot of pictures, JavaScript or other non-text content. The second key aim was to make content understandable and navigable (Chisholm et al., 2001). An example of this can be the use of heading tags to label titles instead of just making the text larger in the style sheet. The WCAG guidelines themselves are a large set of checkpoints that can be used to check conformance. Within these checkpoints there are three priority levels that developers can satisfy (priority 1: ‘must’; priority 2: ‘should’; priority 3: ‘may’). Conformance to these levels is indicated using ‘A’, ‘AA’, or ‘AAA’, where the ‘A’ level is easiest to fulfil and ‘AAA’ is the most difficult (Chisholm et al., 2001).

In 2008 version 2.0 of the WCAG was presented (Caldwell et al., 2008). In this version three important changes were made in comparison to the WCAG 1.0, namely:

- The term ‘web page’ was extended to mean more than just a static HTML page. From then on, it would also cover, for example, dynamic web pages.
- Several criteria were added for content that can be programmatically determined; for example, screen readers should be able to automatically extract certain information.
• Technologies have to be used that are accessibility supported, that is, they should be usable with assistive technologies.

As mentioned previously, the goal of the WCAG is to make web pages more accessible to people with disabilities. For people with a visual impairment this can mean, for example, that pictures have alternative text or that there is a link that allows them to skip over all the navigation links to the main content of a web page.

Even though there are guidelines, being accessible is not a property for websites that developers tend to focus on. Lazar, Dudley-Sponaugle, and Greenidge (2004) performed a survey with 175 webmasters who developed websites in the United States. One of the resulting finds was that only 38.9% had ever tested their web page with a screen reader. Only 24.6% of the webmasters indicated that their websites conformed to the accessibility guidelines issued by the United States Government. Their reasons for not focussing on accessibility more varied from lack of time to the small size of target group. There are few sites that conform to the WCAG and other guidelines, and those that have the logos to show that they do conform might overstate their level of conformance (Petrie, Badani, & Bhalla, 2005).

WAI-ARIA Markup
Another set of guidelines that are concerned with the accessibility of websites are the guidelines for the WAI-ARIA markup. When using this markup so-called ARIA labels can be added to web content to make their processing with a screen reader more accessible (Borodin et al., 2010). They can, for example, be added to dynamic web content to describe its state, property or role. (Hailpern, Reid, Boardman, & Annam, 2009). A simple example of the use of an ARIA label can be the addition of ‘role=link’ to the HTML code for a link.

Other Sets of Guidelines From the Literature
In the literature several other sets of guidelines are suggested that can be used when developing web applications for the blind. There are, for example, the 19 guidelines by Correani et al. (2004), which are grouped in three categories (effectiveness, efficiency, and satisfaction). Another set was suggested by Leuthold et al. (2008). They named 9 guidelines, which were also divided into three categories (context and orientation information, navigation, and clarity and simplicity). Another set of guidelines was that by Hailpern et al. (2009). Theofanos and Redish (2003) also suggested a set of guidelines, which consisted of 32 items. Babu et al. (2010) suggest four main principles (perceivable, operable, understandable, robust) and give us 18 checkpoints to make sure that these main principles are followed.

Of course there are also specific suggestions and guidelines to improve online search for the blind. These will be discussed in the following section.

2.5.2 Guidelines For Online Search
Several papers have been published about the development of search engines for the blind in which suggestions have been made as to what is important in such an interface. The papers will be discussed in more detail in the following subsections, but their guidelines can be generalised to the following groups:

• Clear structure
• Easy navigation
• Clear content and context
The papers also state that the existing guidelines (that is, the sets of web accessibility guidelines named above) should be followed. The groups mentioned above will be discussed more elaborately in the following subsections.

**Clear Structure**
Having a clear structure is a necessity that is mentioned as one of the key features of an accessible website. Therefore the literature proposes guidelines that suggest how to structure the general layout of such pages. For example, ‘simplicity’ (Leporini et al., 2004), ‘place the most important items at the top of the source file’ (Leporini et al., 2008). But they also include specific suggestions for the page with the search results. For example, ‘clearly knowing the number of results obtained’ (Buzzi et al., 2004), ‘arranging the results in numbered lists’ (Andronico & Buzzi, 2005; Leporini et al., 2008), ‘highlighting the results with a heading’ (Leporini et al., 2008), and ‘layout of search results’ (Leporini et al., 2004; Gooda Sahib et al., 2012). Buzzi et al. (2004), Leporini et al. (2004), Andronico, Buzzi, Castillo, and Leporini (2006), and Gooda Sahib et al. (2012) all mention that clustering might permit users to navigate the results more effectively if implemented correctly (that is, accessible and usable). None of these papers mention that they incorporate the clustering of results themselves, but it is not stated why this is not done. Even though Gooda Sahib et al. (2012) state that more research is needed to incorporate the clustering of search results in an accessible search engine they do not mention clustering in their following papers (Gooda Sahib et al., 2013, 2015).

**Easy Navigation**
Easy navigation is important since blind users cannot use their mouse to go from one location on a web page to another. Focus points in this group include, for example, ‘reaching the result area rapidly’ (Buzzi et al., 2004), ‘rapid navigation’, (Andronico & Buzzi, 2005), ‘rapid access’ (Leporini et al., 2004), ‘place the result navigation at the end of the results’ (Leporini et al., 2008), and ‘navigating faster’ (Leporini et al., 2008).

**Clear Content and Context**
Users need to know what the page’s status is and what the function of elements on a page is. For example, ‘alerting by sound’ (Andronico & Buzzi, 2005; Leporini et al., 2008), ‘include auditory previews and overviews’ (Gooda Sahib et al., 2012), ‘labelling fields’ (Leporini et al., 2004, 2008), ‘design interface components that provide the right type of information scent’ (Gooda Sahib et al., 2012), and ‘reading appropriate links about the results’ (Buzzi et al., 2004).

2.5.3 Applicable Guidelines

The sets of guidelines mentioned above already suggest design choices that can be made when designing a website for the blind. The guidelines that are specifically formulated for usable search interfaces for the visually impaired were already discussed as part of the generalization into three groups that was made, but for ease of reference the relevant guidelines can be reviewed in Table 1.
Table 1: An overview of the guidelines for designing online search engines aimed at visually impaired users that are applicable to those that use a screen reader (that is, guidelines on visual representation or creating accessibility for use with, for example, a magnifying glass are outside of the scope). The first column provides the paper that the guidelines were published in, the second column provides the guidelines and columns three to five provide the three groups, that were listed previously, into which the guidelines can be generalized.

<table>
<thead>
<tr>
<th>Source</th>
<th>Guideline</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andronico and Buzzi (2005)</td>
<td>1. Easy location and labeling of edit field and search options.</td>
<td>✓</td>
</tr>
<tr>
<td>Andronico et al. (2006)</td>
<td>2. Highlighting the search result.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>3. Arranging the results in numbered lists.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>4. Recognizing sponsored links.</td>
<td>✓ ✓</td>
</tr>
<tr>
<td></td>
<td>5. Adding navigation and help links.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7. Alerting by sound.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>8. Using standards, such as aural style sheets.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>Leporini et al. (2004)</td>
<td>Simplicity – It is important that the interface be simple.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Labelling fields – Using appropriate tags for assigning a label to the associated field is very important in order for the screen reader to recognize it easily.</td>
<td>✓ ✓</td>
</tr>
<tr>
<td></td>
<td>Rapid access – Navigation and positioning over main fields and results is faster if access keys and tabindex values are defined.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Navigation links – Links or buttons “Next” and “Previous” are particularly important when navigating by keyboard.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Layout of search results – Search results are the basic component of a search engine.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Interaction with search elements. In order to perform a search quickly, it is very important for users to easily identify the search box and buttons, and the configuration options (language, advanced search).</td>
<td>✓ ✓</td>
</tr>
<tr>
<td></td>
<td>Results. Since the main purpose of a search engine is to retrieve information, it is crucial to clearly highlight the search output and make the result more usable.</td>
<td>✓ ✓</td>
</tr>
<tr>
<td></td>
<td>Identification of sponsored links. It is useful to clearly indicate sponsored links for easy visual identification as well as aural recognition.</td>
<td>✓</td>
</tr>
<tr>
<td>Leporini et al. (2008)</td>
<td>1. Place the most important elements of the interface at the top of the source file.</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table continues on the next page...
Table 1: An overview of the guidelines for designing online search engines aimed at visually impaired users that are applicable to those that use a screen reader (that is, guidelines on visual representation or creating accessibility for use with, for example, a magnifying glass are outside of the scope). The first column provides the paper that the guidelines were published in, the second column provides the guidelines and columns three to five provide the three groups, that were listed previously, into which the guidelines can be generalized.

<table>
<thead>
<tr>
<th>Source</th>
<th>Guideline</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leporini et al. (2008)</td>
<td>2. Navigating faster. Assign a scale of importance (i.e., by the tab index attribute) to most important elements, so that users can reach the most important parts quickly.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>3. Alerting by sound.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>4. Using standards, such as XHTML and CSS to separate content from rendering and structure the page.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>5. Easy location and labelling of edit field and search options.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>6. Highlighting the search result.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>7. Arranging the results in numbered lists.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>8. Recognizing sponsored links. Keep sponsored links separate from the other results in the HTML code, not only in the visual layout.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>9. Adding navigation and help links.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>1.2. Number of links and frames.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>1.3. Specific sections.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>1.4. Importance levels of elements.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>1.5. Proper form layout.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>1.6. Assignment of shortcuts.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>2.1. Proper link content.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>2.2. Proper name for frames, tables and images.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>3.1. Messages and dynamic data management.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>3.3. Addition of short sounds.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>4.1. Layout and terminological consistency.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>4.2. Page information.</td>
<td>✓</td>
</tr>
<tr>
<td>Gooda Sahib et al. (2012)</td>
<td>Design interface components that provide the right type of information scent.</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Consider at which stages of the information-seeking process the target group of users are most likely to need support.</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td></td>
<td>Display search results to allow more efficient results exploration.</td>
<td>✓</td>
</tr>
</tbody>
</table>
The guidelines mentioned in the table have some overlap. To reduce this overlap we merge and reformulate these guidelines and group them by the three categories mentioned earlier. As a result we obtain the following set of guidelines:

- **Clear structure**
  - Keep it simple.
  - Use a logical ordering of elements. This helps users in building a mental model of the website and makes it easier for them to predict where they might find the information they need.
  - Use the correct HTML and WAI-ARIA tags for the items on a webpage. For example, use header tags for headings and use tags such as ‘role=search’ on the search field.
  - Use lists to present the search results.
  - Make the search field and button easy to find.
  - Order page elements by importance.
  - Be consistent across pages in structure and naming.

- **Easy navigation**
  - Provide links for navigation when needed and provide a link to a page where users can find help information.
  - Where possible, provide shortcuts for fast navigation, as these can be useful.

- **Clear content and context**
  - Avoid inaccessible elements. Elements such as dynamic suggestions are difficult to use for blind users.
  - Use short auditory notifications. An auditory notification that the page has loaded successfully or that something went wrong provides the user with an extra cue.
  - Do not put advertisements or sponsored links in the same list as the results, and make it known to the user that they are advertisements.
  - Provide clear alternative descriptions for elements such as images.

2.6 Search Engine Implementations

As discussed in the previous sections there are many guidelines that a search engine implementation for the blind could (or should) conform to. There have been two research groups that have developed such search engines. Both groups have published a series of papers through which their design process and considerations can be followed.

The first group is that of Buzzi et al.. In Buzzi et al. (2004) they state that they had tested the accessibility of seven search engine interfaces and that at that time only Google conformed to the A level of the WCAG 1.1. They also found that 92% of sighted users found search engines easy to use, but only 7% of blind users agreed. They conclude that redesigning existing search engine interfaces to be more accessible would have a low cost and large benefits for the target group. In a second paper (Leporini et al., 2004) they propose guidelines for developing search engine interfaces. They also give a more elaborate overview of the differences between conventional navigation and navigating using a screen reader, and the problems that screen reader users encounter. Finally, they again stress how search engine companies could attract a greater number of users if they would modify their layouts to be more accessible. In a paper by Andronico and Buzzi (2005) the authors summarize the previous papers (problems encountered, suggested guidelines) and describe how a screen reader (in this case JAWS (Freedom Scientific, 2015)) processes a web page. They then continue to describe two scenarios that they used for the development of their search engine
implementation. The first scenario is that of a blind user interacting with the original Google interface and the second scenario is that of a blind user interacting with their proposed modified Google interface. In the conclusion of this paper they state that their next step would be to test an implemented version of their proposed application. In Andronico et al. (2006) all the information from previous papers is summarized (graphs from the survey, problems encountered, suggested guidelines, how JAWS processes a web page). They then proceed to describe how they created their implementation of an accessible version of the Google search engine. They key adaptations they make are:

- Re-writing the code to standards. That is, create clear sections and use proper HTML tags.
- A new logical order. That is, change the ordering of the clear sections in the source code.
- Additional features. That is, enable quick navigation through short keys, add hidden labels, and add aural feedback (notification sounds).

To enable their own implementation they used the Google API to retrieve the search results and then displayed these results a the web page with their restructured layout. One of the limitations of this implementation was that they had finetuned it for usage with JAWS, which is a commercial screen reader that at that time only worked in Internet Explorer\(^2\). In the 2008 paper by Leporini et al., the development steps previously mentioned are summarized and the testing process of the developed interface is also described. The modified Google interface was remotely tested by 12 blind users. Remote testing was chosen because the environment in which they use a web page can be of great influence on the results that can be obtained from a blind participant. For example, blind users are accustomed to their own operation system, web browser and screen reader. This means that placing them behind a different computer (and keyboard), having them use a different web browser, or use a different screen reader might introduce many new elements that provide them with an extra learning curve or possible extra obstacles. By remote testing, users could use their own set up. The result of the evaluation was that the tests with the 12 participants confirmed the value of the design guidelines that had been followed.

The second research group was that of Gooda Sahib et al.. They started with a study investigating the information-seeking behaviour of visually impaired and sighted searchers, and how these two groups compare (Gooda Sahib et al., 2012). They performed an observational study with 15 visually impaired (13 of which had no vision, and 2 of which had very low-level vision) and 15 sighted participants. The visually impaired participants were allowed to use their own computers (and therefore operating systems/browsers/screen readers). They found that visually impaired users submitted slightly longer queries on average, they navigated the search results page in a header-to-header manner (or link-to-link). Sighted users more often visited multiple search results pages (43% against 13%) and viewed more search results (13.40 on average against 4.27 on average). They also describe that sighted users could ‘take a glance’ at the results after searching and quickly adjust their query if they needed to. Sighted participant also submitted more queries (10.93 on average against 4.47 on average). They conclude their first article by suggesting guidelines and state that they will have to design and implement a search interface to be able to user test evaluate their guidelines. Their second paper (Gooda Sahib et al., 2013) describes the process of participatory design that they apply to design their search interface. This participatory design process has two types of potential user involvement, that is, they consult with a blind team member during development and they ask four blind users for their input during prototyping sessions. They conclude that this is a promising design method, but that it should be kept in mind that the preferences and advice of a single blind team member might not be universal for all blind users. In

\(^2\)It is currently still limited to Windows.
their last paper (Gooda Sahib et al., 2015) they describe, TrailNote, the search engine they have developed and the process of its evaluation. TrailNote also makes use of the Google Search API to obtain the results, saves the user’s queries and selected results in a ‘search trail’ and allows them to tag results as useful. The search engine also as a note taking area. A user can also select options when interacting with the search results such as, for example, ‘save to note’. The evaluation of TrailNote consisted of two sessions in which participants were asked to perform search tasks (each participant participated in two sessions). The first session for each participant took place remotely over Skype to let the participant use their own system and screen reader. Participants first filled in a concession and a demographic form, and were then given a training task. This was followed by an explanation about the testing task, which they then performed for half an hour after which they were asked to fill in an evaluation form about the system’s features. Information on the participants’ interaction was gathered through screen recording software and system logging, and a semistructured interview was conducted to gain insight in the partipant’s experience and to obtain feedback on the overall impression. In the second session the saved search trail and notes from the participant were loaded and they were explained their new search task. The participant was also asked about their strategies on how they would search for previously encountered information and resume their previous search. They then again performed their task for half an hour while information was gathered through screen recording software and system logging. Finally they were given a short questionnaire about the interface and again a semistructured interview was performed to evaluate their experience with the new task and refinding information. The conclusion by Gooda Sahib et al. (2015) was that their participants seemed to like the system. They also noted that they, as concluded in Section 2.2, found usability very important in addition to accessibility. They also stated that it was important to have potential users included in the design process. Finally they name a redesign of dynamic suggestions or more research into sound notifications as possible options for future research.

2.7 Summary

As stated in Section 2.3 there are four groups of problems that the blind can encounter when they browse the internet, that is, lack of context, sequential navigation, cognitive load, and screen reader interpretation. There are methods to improve the usability of websites for blind users by limiting the occurrence of these problems. However, as stated in Section 2.2 the construction of a website should be carefully thought through to make sure that it is not only accessible, but also usable. When constructing a website, one can begin with learning about often occurring problems, and which browsing strategies and coping behaviour the blind apply (Section 2.4). This can then be followed by using guidelines and suggestions from the literature that provide advice on how to design a search engine for the blind (Section 2.5.3) during the design and implementation phases. These are an addition to the more general guidelines for designing websites for the visually impaired (Section 2.5). Therefore, we can conclude that, in order to develop a website that is accessible and usable, conformance to the guidelines mentioned can be seen as a requirement when developing a search engine for the blind.
3 Design

The development of the application started with listing the scope, that is, what is the target group, which type of application do we want, and what search results do we want to present to the user. Then the elements that should be a part of the application were defined. This was followed by taking the guidelines listed in the previous chapter (Section 2.5 and Section 2.5.3) and formulating additional guidelines where needed. When these guidelines combined with the requirements set by the definition of the scope this resulted in the final set of requirements. The final step in the design process was the design of the content of the web search interface and applying the guidelines in that design.

3.1 Scope

The first step was to define the scope for the project. This was done by defining the target group, deciding on the type of application and defining the type of search results to be presented to the user.

3.1.1 Target Group

As mentioned in the introduction, in this thesis, we distinguish three groups with different levels of sightedness, namely: the blind, the partially sighted and the sighted. It was decided that the target group for the application would be the blind. The variability in levels of sightedness in the partially sighted group is very large ranging from nearly blind to nearly sighted, and therefore they also use a wide range of aids. Some visually impaired users use a screen reader (as the blind do), but others just use a magnifying glass or enhanced contrast. We therefore decided that the application would be developed with a focus on usability with screen readers. Additions to the application to improve it for use with magnifying glasses and/or enhanced contract were saved for possible future research. There was no strict specification in terms of the age or internet skills of targeted users, but the intention was that the application should be usable for users across a broad range of skill levels. This meant that the website would have to be easy to navigate for everyone that is able to locate the browser window on their computer with a screen reader. The navigational skills with a screen reader could range from the level of a user of Microsoft Narrator who only knows how to press the Tab and arrow keys to navigate, up and to the level of an advanced JAWS user who knows how to search for items on a page and use all kinds of key combinations.

3.1.2 Application Type

Since there can be quite a learning curve in using a screen reader - even if you are used to a different one already - it was decided that the application to be developed would have to be web browser and screen reader independent. That is, it should be possible for users to use their preferred operating system, browser and screen reader. Therefore, the application was decided to be a web application. A web application, as opposed to a desktop application, would not require users to install any extra software and such a page would be accessible on all platforms (Windows, Linux and OS X) - assuming that the user’s preferred screen reader, a web browser and a working internet connection are present.
3.1.3 Search Results

The task of a search engine is to present the user with promising links that might contain the information they seek. The current state-of-the-art search engine can also be used to search for images and videos. In the case of the search interface to be designed this information will be restricted to the presentation of search results in the form of textual results (that is, links to web pages with a descriptive text). To make videos or images accessible for blind users, additional detailed descriptions of their content would have to be made available - either made by humans or computers, but an added layer of information none the less. For this project, the scope was therefore limited to links and their descriptions.

3.2 Minimal Application Features

To be usable as a search engine the application should be able to take a user’s query and present them with the found search results. An admin panel would also be useful to effectively maintain the application and to be able to make small changes to its settings without having to edit the source code and restart the application.

3.3 Requirements

Formulating the requirements was the second step in the design process. These requirements were a combination of the requirements that follow from the definition of the scope and minimal application features, the general guidelines for accessible web development (that is, the WCAG and WAI-ARIA guidelines we discussed in Section 2.5), and the applicable guidelines listed in Section 2.5.3.

The requirements that follow from the definition of the scope and minimal application features were that the application should:

- Not only be accessible but also usable by the target group (the blind).
- Be usable with a screen reader.
- Be usable for users ranging from a ‘beginner’ level of computer skills to ‘advanced’ users.
- Be a web browser independent web application.
- Be screen reader independent.
- Provide the users with textual search results (that is, links with a descriptive text).
- Let a user ask their search question.
- Present a user with search results.
- Have an admin panel.

The conformance to the general web accessibility guidelines listed in Section 2.5, that is, the WCAG and WAI-ARIA guidelines, was set as an requirement as well. Conformance to the applicable more specific guidelines listed in Section 2.5.3 was also added to the list.

We also defined some features that we would like the application to have. These were:

- The search results would be presented as clusters.
- A button that allows users to retrieve more search results for a cluster category.
- An autofocus on the search question input field.
- An automatic focus on the results list when loading the results page.

These features were preferred since it became apparent from the literature research that the key feature of usable websites for the blind was that they were well structured, intuitive and consistent, which made them easy to navigate. A website that fulfils the requirements we listed and has
these features has a chance of preventing problems that fall into the four groups of general website problems (lack of context, sequential navigation, cognitive load, screen reader interpretation). As was briefly named in the papers by the research groups of Buzzi et al. (2004) and (Gooda Sahib et al., 2012) a possible manner of introducing extra structure in the presentation of the search results is clustering (although they did not implement this or mention why they did not do so). As we agree with them that clustering could provide an improvement in the presentation structure for search results we decided that clustered results would be a feature that we wanted to implement. The button that allows users to obtain more search results in a cluster category is useful since they would not have to navigate back to the search field when wanting to retrieve more results. The autofocus on the search field and focus on the results list features will also make sure that users do not have to search for these elements when the page is loaded, which also improves the ease of navigating the website.

With our requirements defined, the design of the web search interface’s content was started.

3.4 Content Design

After having formulated the minimal application features and the requirements it was time to design the content of the web application. That is, which pages should there be and what should be on them. This will be described in the following subsections. After that description we will describe what the search result information is that we want to present.

3.4.1 Page Content Design

Since it was decided that the search engine’s pages would ideally only contain necessary content, we specified that our web search interface would need something that allows the user to specify their search query and the possibility to view the results. It was therefore decided that there would be a search page and a results page. In the following subsections a short overview of the needed elements on these pages is given. An element is a ‘needed element’ when it is required for the website to function as a search engine. Also, since an admin panel is preferred, an overview of the elements needed in an admin panel will be given, even though unprivileged users will not have access to this page. Of course the elements and the web pages themselves will need to follow the structural guidelines (such as the WCAG) that were summarized in the background chapter, but those details will be described in the chapter about implementation.

The Search Page
The necessary basic elements that the target group needs for a search page are the following:

- A search field, which allows the user to type in a query.
- A search button, which allows the user to submit that query.
- A help link to a page that provides the user with some short information.

The Results Page
A results page also has a necessary element, which is the following:

- A presentation of the results.

The Admin Panel
The admin panel should contain settings that would enable the owner of the search engine to tune its settings. The following elements would be needed:

- A means of authentication, so that only those that are authorized can change settings.
• A page for changing the layout settings.
• A page for changing the settings for retrieving the results.
• A page for changing the cluster algorithm settings.

3.4.2 Search Results

To obtain the results that fit a user’s search question (or query) the results of one of the state-of-the-art search engines will be used. This choice was made for both ethical and practical reasons. Firstly, the goal was to improve the search experience of visually impaired users. They should, however, have access to the same information that sighted users are presented with. Therefore, using the results from one of the popular search engines, would present the visually impaired user with equal information as when they would use that popular search engine. Secondly, deciding which search results to present a user with has been an extensive topic of research. It requires scraping the web and saving large amounts of data. It also requires complex algorithms to rank the found results based on estimated user preferences. Therefore, it makes more sense to use the results returned by the state-of-the-art search engines. This also means that the developed application stays up to date in terms of ranked results, since any improvements made to algorithms on the supplier’s side are automatically included.

The search results that are obtained at least contain the following information per item (independent of which state-of-the-art supplier is used):

• **Title.** The title of the result. Usually identical to the title of the web page to which is being referred.
• **URL.** The link to the web page.
• **Snippet.** A short description that aims to provide the user an idea of the content of a web page. This can be the web page’s description, if present, but it might also be a part of the paragraph that the search terms were found in.

An example of how a search result usually is displayed in Bing and Google can be found in Figure 7a and Figure 7b, respectively. Which state-of-the-art supplier is chosen will be described in the chapter about implementation.

![Bing Example](image)

(a) Bing

![Google Example](image)

(b) Google

Figure 7: An example of what the same search result for the ‘information’ page on Wikipedia looks like in Bing and Google, respectively.
3.4.3 Clustering

As stated above in the requirements (Section 3.3) and suggested in the papers by the research groups of Buzzi et al. (2004) and Gooda Sahib et al. (2012), the structure of the results page might be improved by clustering the results. A user could get a general idea of what is on the results page without having to visit every search result. That is, a user can look at the a cluster’s name and if it is not the sought for subtopic they are able to skip multiple results at once. This could speed up the search process considerably because a user would be able to reach the promising results faster or would be able to conclude faster that the search question has to be adjusted. To be able to present the user with the clustered search results, the results have to be clustered. This means that an algorithm is required that is able to cluster small amounts of text. The choice for such an algorithm and the internal functioning of the chosen algorithm is the subject of the following chapter.
4 Clustering

As stated in the definition of the requirements in the previous chapter, the search results would be clustered to present them in a structured manner. This meant that a clustering algorithm had to be applied. The search results that such an algorithm should cluster each consist of at least the following three components:

- **Title.** The title for the search result. This is usually the name of the website.
- **URL.** The link to the website.
- **Snippet.** A short description that aims to provide the user an idea of the content of a website. This can be the website’s description, if present, but it might also be a part of the paragraph that the search terms were found in.

The number of search results that are usually used in search results clustering tasks is 50-100 results for a single query. This might seem like a small amount, but it corresponds to 5 to 10 pages of results when displayed by a search engine, and - to be honest - most search engine users do not look beyond the first page. Before the choice was made for an algorithm some requirements were defined, which will be described in the following section.

4.1 Algorithm Requirements

Given the type of data and the target group there were several requirements that should be followed when choosing the algorithm. These requirements were the following. Firstly, the algorithm should be able to handle the type of information provided by search results, which means that each item only consists of a limited amount of text (namely, a title and a short snippet). Secondly, it should be able to provide clusters based on about 50 or perhaps 100 of these short-text search results. As mentioned above, this is the amount of search results that are most commonly used with search results algorithms. Using less results would make it hard to cluster the snippets correctly, but using more could introduce irrelevant search results into the data set and thus result in irrelevant cluster categories. Thirdly, the algorithm would need to provide a high precision in assigning search results to the clusters created. That is, having a good retrieval rate would also be important, but the possible confusion that impure clusters, that is, clusters that contain off-topic results, would have to be prevented. Fourthly, it would be preferable to use an algorithm that would be able to do so in an nearly instantaneous manner. This would be preferable because there is a huge amount of information on the internet and many examples would be needed if a reliable model would have to be build. Also, such a model would have to be kept up to date. Fifthly, the algorithm had to automatically provide names for the clusters it had created and these clusters and names would need to have a meaningful cohesion for the user. Sixthly, the proof of concept was to be implemented for Dutch users, but it is not uncommon for Dutch users to also provide queries that are in English, which would mean that the algorithm had to have no problems with multiple languages being used. To summarize, the chosen algorithm has at least be able to:

- Work on snippets
- Work on few items
- Have a high precision
- Has to work nearly instantaneous
- Provide meaningful names (or be ‘description centric’)
- Be multilingual

A few preferences can be added to these main requirements. These are:
• Stop words should remain intact
• No information base needed
• No training is needed

The first preference ‘stop words intact’, has been formulated since one of the methods to reduce complexity in processing text is to remove often occurring words such as stop words from the data (van Rijsbergen, 1979). Since we would like to have cluster labels that are as complete as possible and provide a clear description, using an algorithm that does not exclude stop words would be nice to have. The reason for the second preference, ‘no information base needed’, is that there are clustering algorithms in which an information base models of concepts are used. Such an information base can, for example, be built by using the information represented in WordNet (Miller, 1995) or as has been a more recent development by building a concept structure based on the links between Wikipedia (Foundation, 2017) pages. As can be imagined, building such a information base can take a lot of time and resources, and there is a possibility that there are concepts that are not included. Having an information base also makes the ‘multilingual’ requirement difficult to fulfill since a base should then be built for each language that has to be included. The reasons for the ‘no training needed’ preference overlap mostly with the reasons for not wanting an information base. Training a model means that the data for training that model has to be as complete as possible and that the model should be able to process multiple languages or that multiple models should be trained.

4.2 Clustering Algorithms

In the literature multiple options for clustering algorithms can be found. When specifying the category of clustering as methods operating on small pieces of text (that is, the titles and snippets from search results) there are a number of methods for solving that task can be found. To get a reasonable overview of the developments in the field one could start by reading the survey by Carpineto, Osiński, Romano, and Weiss (2009). Secondly, to get an overview of more recent developments the paper by Cobos et al. (2014) provides a detailed overview in their background section. An overview that we created based on the information provided in their descriptive papers of search results clustering (SRC) algorithms and their conformance to the requirements set above can be found in Table 2. Based on these conformances, the description-centric algorithm Lingo (Osinski, 2003; Osinski, Stefanowski, & Weiss, 2004; Osiński & Weiss, 2005) was chosen to cluster the search results for the search interface.

4.3 Lingo - How Does It Work?

The description-comes-first algorithm Lingo clusters search results. In the following subsections the steps it takes to do so will be described. The working of the algorithm is also described in multiple papers by its developers (Osinski, 2003; Osinski et al., 2004; Osiński & Weiss, 2005). The algorithm starts its clustering process as follows: There is an original list of search results that you would like to cluster (say 50 or 100 results that have the Title, URL, and Snippet format described previously). In order to cluster these results a few steps will be taken. In short:

• Step 1: Preprocessing
• Step 2: Phrase extraction
• Step 3: Cluster label induction
• Step 4: Cluster-content allocation
• Step 5: Final cluster formation
Techniques that are applied in Step 1 include: text segmentation, stemming, and stop word marking. Step 2 contains the use of Suffix Arrays and Longest Common Prefix (LCP) arrays. Step 3 involves the building of a term-document matrix (using tf-idf) and applying Latent Semantic Indexing (LSI) using Singular Value Decomposition (SVD). The use of LSI enables the algorithm to discover 'abstract concepts', that is, the algorithm can discover and deal with, for example, synonyms. Steps 4 and 5 involve simple matrix multiplication, and cluster scoring and sorting. The following subsections will each further describe how these listed steps work (based on the information provided in the papers (Osinski, 2003; Osinski et al., 2004; Osinski & Weiss, 2005)). Pseudo-code for the algorithm can be found in Algorithm 1 in Osinski et al. (2004, p. 4). The relevant pseudo-code is included below for each step for ease of reference.

### 4.3.1 Step 1: Pre-processing

The first step in clustering the search results is to pre-process the snippets and titles. When starting with the raw search result data the data first has to be preprocessed. For Lingo that means that it applies three common text processing techniques: text-segmentation, stemming, and marking and ignoring stop words. This combination of techniques is one that is quite standard for text processing (a process already described by van Rijsbergen (1979) although in Lingo the stop words are marked and not entirely removed.) The text-segmentation methods are applied first and are a means of dividing the snippets and titles into tokens. During the division process items such as
Algorithm 1: Pseudo-code for Step 1: Pre-processing. Taken from Osinski et al. (2004, p. 4). $D$ resembles the collection of snippets that are the input for the algorithm.

```plaintext
{Step 1: Preprocessing}
for all $d \in D$ do
    perform text segmentation of $d$; 
    {Detect word boundaries etc.}
    if language of $d$ recognized then
        apply stemming and mark stop-words in $d$;
    end if
end for
```

HTML tags and non-letter characters are removed. Lingo then tries to recognize the language for each search result before applying stemming and stop word marking. If the algorithm recognizes the language then stemming and stop word marking are applied. The language is recognized by using stop words as indicators, that is, the assumption is made that the language for which the most stop words are recognized in the search result is the language in which the search result is written. Stemming is used to transform words into their word stem (or lemma). This reduces variability in the dataset, for example, ‘talking’, ‘talks’, and ‘talked’ all become ‘talk’, which allows them to be recognised as the same word. The original form of the stemmed words is stored as well since these forms might be needed when these words are selected as part of the cluster labels. The second technique, the marking and ignoring of stop words, is used to not have function words as, for example, ‘the’, ‘by’ and ‘to’ influence the clusters by seeming important since they occur very frequently. As with the stemmed words, the marked stop words are not completely removed since they might be important for the readability of the cluster labels that are constructed in later steps. The pseudo-code for this step can be found in Algorithm 1.

4.3.2 Step 2: Phrase Extraction

Algorithm 2: Pseudo-code for Step 2: Phrase extraction. Taken from Osinski et al. (2004, p. 4). Note: ‘documents’ refers to the pre-processed snippets.

```plaintext
{Step 2: Frequent Phrase Extraction}
concatenate all documents;
$P_c \leftarrow$ discover complete phrases; $P_f \leftarrow p$: 
{ $p \in P_c \land \text{frequency}(p) > \text{Term Frequency Threshold}$ }
```

The second step is the extraction of frequent phrases. The developers of the algorithm describe frequent phrases as ‘recurring ordered sequences of terms appearing in the input documents’. Phrases that occur more often in the results should indicate a common topic and can therefore be used to determine the names of clusters. Of course, synonyms might be used while talking about a topic, but the authors believe that Lingo can partially overcome that problem by applying the following steps when determining the frequent phrases. There are four requirements for a frequent phrase:

1. It should appear often enough in the search results.
2. It should not cross sentence boundaries.
3. It should be a ‘complete phrase’.
4. It should not begin or end with a stop word.
A ‘complete phrase’, as mentioned in these requirements is the ‘longest possible’ frequent phrase (for example: ‘search results clustering algorithm’ instead of just ‘results clustering’). To find the complete phrases that can be used as potential cluster labels first a suffix array is built. This is done by taking a long text that consists of all the search result texts concatenated and treating the words as the smallest units that the text consists of. That is, the smallest form for a suffix is a word. An LCP array is then built from the suffix array. An example using characters instead of words as smallest units can be given for the sentence $S = \text{banana}\$ (where the $\$ \symbol$ is used to indicate the end of the sentence). The indexes for the characters in the string are then:

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S[i]$</td>
<td>b</td>
<td>a</td>
<td>n</td>
<td>a</td>
<td>n</td>
<td>a</td>
<td>$$</td>
</tr>
</tbody>
</table>

The suffixes that can be found for the string are the following (in their initial order and sorted):

<table>
<thead>
<tr>
<th>Suffix</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>banana$</td>
<td>7</td>
</tr>
<tr>
<td>anana$</td>
<td>6</td>
</tr>
<tr>
<td>nana$</td>
<td>4</td>
</tr>
<tr>
<td>ana$</td>
<td>2</td>
</tr>
<tr>
<td>na$</td>
<td>1</td>
</tr>
<tr>
<td>a$</td>
<td>5</td>
</tr>
<tr>
<td>nana$</td>
<td>3</td>
</tr>
</tbody>
</table>

The suffix array for this sentence is then (which corresponds to the sorted suffixes in the previous table):

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A[i]$</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

The LCP array that can be build from this suffix array is then:

<table>
<thead>
<tr>
<th>i</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H[i]$</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Which means that the longest common prefix for suffix A[3] and A[4] has length 3 ($\text{ana}$ for $\text{ana}\$ and $\text{anana}\$).

In the Lingo algorithm the suffix arrays are built with words as their components (instead of the characters in the $\text{banana}\$ example) and using the LCP array often occurring prefixes are found (which in this case are phrases). The algorithm then finds the longest versions of these prefixes. If it encounters prefixes that are longer (but overlapping) than the one on top of the stack the shorter prefix is dropped. If the current prefix is equal to the one on top of the stack its frequency is increased. If the current prefix is shorter than the one on top of the stack the prefix on top of the stack is output as a right-complete substring. To find the complete substrings they must be left and right complete. First the right-complete phrases are discovered as described. Then the left-complete phrases are discovered by applying the same algorithm on the reversed text. Then the results for the left-complete phrases are sorted alphabetically. Next the right- and left-complete sets are combined to get the set of complete phrases. Finally, a last selection is performed by keeping the phrases that exceed the Term Frequency Threshold. The pseudo-code for this step can be found in Algorithm 2.

### 4.3.3 Step 3: Cluster Label Induction

This step consists of two parts. Part 1 consists of building the term-document matrix (matrix $A$). A term in this case is a word and a document is a search result. Terms that were marked
Algorithm 3: Pseudo-code for Step 3: Cluster label induction. Taken from Osinski et al. (2004, p. 4). $P_f$ is the set of frequent phrases resulting from Step 2.

\begin{algorithm}
\{STEP 3: Cluster Label Induction\}
\begin{itemize}
    \item $A \leftarrow$ term-document matrix of terms not marked as stop-words and with frequency higher than the Term Frequency Threshold;
    \item $\Sigma, U, V \leftarrow$ SVD($A$); \{Product of SVD decomposition of $A$\}
    \item $k \leftarrow 0$; \{Start with zero clusters\}
    \item $n \leftarrow \text{rank}(A)$;
    \item \textbf{repeat}
        \begin{itemize}
            \item $k \leftarrow k + 1$;
            \item $q \leftarrow (\sum_{i=1}^{k} \sum_{ii})/(\sum_{i=1}^{n} \sum_{ii})$;
        \end{itemize}
    \item \textbf{until} $q < \text{Candidate Label Threshold}$;
    \item $P \leftarrow$ phrase matrix for $P_f$;
    \item \textbf{for all} columns of $U_T^t P$ \textbf{do}
        \begin{itemize}
            \item find the largest component $m_i$ in the column;
            \item add the corresponding phrase to the Cluster Label Candidates set;
            \item $\text{labelScore} \leftarrow m_i$;
        \end{itemize}
    \item \textbf{end for}
    \item calculate cosine similarities between all pairs of candidate labels;
    \item identify groups of labels that exceed the Label Similarity Threshold;
    \item \textbf{for all} groups of similar labels \textbf{do}
        \begin{itemize}
            \item select one label with the highest score;
        \end{itemize}
    \item \textbf{end for}
\end{itemize}
\end{algorithm}
as stop words or did not exceed the Term Frequency Threshold value in the previous steps will not be included. The matrix will be built using the tf-idf scheme (Salton & Yang, 1973). That is, \( a_{ij} \) in the matrix will indicate the degree of relationship between the term \( i \) and document \( j \). The value of \( a_{ij} \) is calculated using the \( a_{ij} = tf_{ij} \cdot \log(N/df_i) \) formula. In words: the degree of relationship between the term and the document is equal to the number of occurrences of that term in the document multiplied by the log of the total number of documents divided by the number of documents that the term appears in. By using this formula to calculate the \( a_{ij} \) values terms that occur in many documents, and therefore do not provide much information about possible clusters, have less weight than when only the number of occurrences in the documents is used. More weight is also given to terms that appear in the titles of search results (by multiplying them with the Title Word Boost value). An example of a term-document matrix (with the corresponding terms, documents and phrases) is provided in Figure 8.

![Term-document matrix example](image)

Figure 8: An example of terms, documents, phrases and the term-document matrix that corresponds to them. Taken from Osinski (2003, p.47).

Part 2 is about the discovery of abstract concepts and consists of finding the abstract concepts and matching phrases to them. The abstract concepts are found by performing a Singular Value Decomposition (SVD) on the term-document matrix (again, matrix \( A \)). This means that matrices \( U \), \( \Sigma \), and \( V \) will be calculated so that \( A = U \Sigma V^T \). The first \( k \) columns in matrix \( U \) represent the abstract concepts that can be found in, in this case, the search results. (For an example that follows on the the term-document matrix given in the previous example see Figure 9.) The Candidate Label Threshold value will influence the value of \( k \). In the example the first two columns of matrix \( U \) are selected.

The matching of the phrases is done by creating matrix \( P \) (the cluster-label-candidate matrix). Matrix \( P \) has a row for each of the terms (T1 up and until T5) and a column for the phrases (P1 and P2) followed by again the terms (T1-T5). The values in \( P \) are again tf-idf weighted and normalized for the length of the phrases. The description labels for the clusters are then found by creating matrix \( M \) (the abstract-concept-cluster-label-candidate matrix) by multiplying \( U_k^T P \).
Figure 9: An example of the matrices in the label induction phase. $U$ is the abstract concepts matrix, $P$ is the cluster-label-candidate matrix, and $M$ is the abstract-concept-cluster-label-candidate matrix. Taken from Osiński and Weiss (2005, p.51).

This means that the first two columns of $U$ (that give us two abstract concepts) are combined with the terms and phrases that are in matrix $P$ and thus give us matrix $M$ with the best labels (on the columns - either a term or phrase) for the abstract concepts (on the rows). In the example, we can see in matrix $M$ that for the first abstract concept phrase P1 is the best fit (‘Singular Value’ with a 0.92 value) and phrase P2 fits the second concept best (‘Information Retrieval’ with a 0.97 value).

If multiple cluster labels are found that have a cosine similarity higher than the Label Similarity Threshold only the label with the highest score will be kept. The pseudo-code for this step can be found in Algorithm 3.

4.3.4 Step 4: Cluster-Content Allocation


\[
\begin{align*}
\text{Step 4: Cluster Content Discovery} \\
\text{for all } L \in \text{Cluster Label Candidates do} \\
&\quad \text{create cluster } C \text{ described with } L; \\
&\quad \text{add to } C \text{ all documents whose similarity} \\
&\quad \text{to } C \text{ exceeds the Snippet Assignment Threshold;} \\
\text{end for} \\
&\quad \text{put all unassigned documents in the “Others” group;} \\
\end{align*}
\]

To find the content for each cluster the columns from matrix $P$ that represent the chosen cluster labels (in this case the first and second column) are combined into a new matrix (matrix $Q$). Matrix $Q$ is then multiplied with matrix $A$ to create matrix $C$ ($C = Q^TA$). Matrix $C$ has the cluster labels as its rows and the documents as its columns. If the value in $C$ is above the Snippet
Assignment Threshold the document is added to the corresponding cluster. Documents can be added to multiple clusters, but if they do not reach the threshold for any of the clusters they are added to the ‘Others’ cluster. (For an example that follows on the previous two examples, provided in Figures 8 and 9, see Figure 10.) The pseudo-code for this step can be found in Algorithm 4.

\[
Q = \begin{pmatrix}
0.00 & 0.56 \\
0.71 & 0.00 \\
0.71 & 0.00 \\
0.00 & 0.00 \\
0.00 & 0.83
\end{pmatrix}
\]

\[
C = Q^T A = \begin{pmatrix}
0.69 & 1.00 & 0.00 & 0.00 & 1.00 & 0.00 \\
0.00 & 0.00 & 1.00 & 1.00 & 0.00 & 0.00
\end{pmatrix}
\]

**Information Retrieval [1.0]**
- Introduction to Modern Information Retrieval [1.00]
- Using Linear Algebra for Intelligent Information Retrieval [1.00]
- Automatic Information Organization [0.56]

**Singular Value [0.95]**
- Software Library for the Sparse Singular Value Decomposition [1.00]
- Singular Value Analysis of Cryptograms [1.00]
- Large Scale Singular Value Computations [0.69]

**Other Topics**
- Matrix Computations

Figure 10: An example of the resulting clusters and their content allocated content. Taken from Osinski (2003, p.51).

### 4.3.5 Step 5: Final Cluster Formation

**Algorithm 5:** Pseudo-code for Step 5: Final cluster formation. Taken from Osinski et al. (2004, p. 4).

```plaintext
{Step 5: Final Cluster Formation }
for all clusters do
    clusterScore ← labelScore × ||C||;
end for
```

As a final step the clusters are given a cluster score (which is a multiplication of their label-score and their member-count) and are sorted from highest to lowest score. The pseudo-code for this step can be found in Algorithm 5.

### 4.4 Resorting of Clusters

The state-of-the-art search engines, from which we plan to retrieve the search results, return the results in a certain order. The algorithms behind this ordering provide the user with a list of diverse search results ordered by relevance. The ordering of the clusters that are returned by Lingo is not the same. To make use of the information contained in the ordering by the search engine, the final set of clusters is sorted based on the initial order of occurrence of the search results they contain.

For example, let us take an initial ordering of search results: 1(A), 2(A), 3(B), 4(C), 5(A), 6(B), 7(A), 8(C), 9(B) (where A, B and C represent three different subtopics that can be found in the results). Lingo will then create three clusters: Cluster ’A’: \{1, 2, 5, 7\}, cluster ’B’ \{3, 6, 9\} and
cluster 'C'\{4,8\}. But it could happen that the final ordering of those clusters on cluster score is: C, A, B. To provide users the results that were presented by the search engine as most relevant first the clusters are reordered based on the number of their first search result. Therefore, the ordering will be: A (first result: 1), B (first result: 3), C (first result: 4).

Two examples using actual produced cluster labels can be the following. The first set of labels are returned when the algorithm is set to provide a larger number of clusters, the second is when the algorithm has stricter settings. For both examples the query that was asked was ‘trein Londen’ ('train London'). For the first case this resulted in the following clusters with their translation and the rank of their first result in brackets behind the label:

1. ‘Vervoer in Londen’ ('Transport in London', rank: 20)
2. ‘Stedentrip Londen’ ('Citytrip London', rank: 3)
3. ‘Comfortabel met de Trein naar Londen’ ('With the train to London in comfort', rank: 0)
4. ‘Londen te Reizen’ ('To travel to London', 6)
5. ‘Eurostar Trein’ ('Eurostar Train', rank: 0)
6. ‘Reis met de trein’ ('Travel by train', rank: 0)
7. ‘Informatie’ ('Information', rank: 25), and ‘Other Results’ (which is always the last category)

When the clusters are restructured based on the original ranking of their first results their ordering is:

1. ‘Comfortabel met de Trein naar Londen’ ('With the train to London in comfort', rank: 0)
2. ‘Eurostar Trein’ ('Eurostar Train', rank: 0)
3. ‘Reis met de trein’ ('Travel by train', rank: 0)
4. ‘Stedentrip Londen’ ('Citytrip London', rank: 3)
5. ‘Londen te Reizen’ ('To travel to London', 6)
6. ‘Vervoer in Londen’ ('Transport in London', rank: 20)
7. ‘Informatie’ ('Information', rank: 25)
8. ‘Overige’ ('Other Results', which is always the last category).

For the second case the returned labels are:

1. ‘Stedentrip Londen’ ('Citytrip London', rank: 5)
2. ‘Hotel in Londen’ ('Hotel in London', rank: 5)
3. ‘Trein naar Londen’ ('Train to London', rank: 0)
4. ‘Eurostar’ ('Eurostar', rank: 0)
5. ‘Overige’ ('Other Results', which is always the last category)

When these are restructured based on the orginial ranking of their first results the ordering becomes:

1. ‘Trein naar Londen’ ('Train to London', rank: 0)
2. ‘Eurostar’ ('Eurostar', rank: 0)
3. ‘Stedentrip Londen’ ('Citytrip London', rank: 5)
4. ‘Hotel in Londen’ ('Hotel in London', rank: 5)
5. ‘Overige’ ('Other Results', which is always the last category)

In both cases the most relevant cluster labels (for example, ‘Eurostar’) are moved to the beginning of the list and less relevant labels are moved more towards the end of the list.

Now that the clustering method for the search results was chosen and its workings were understood, the next step in the development process was the implementation of the search interface and tuning it to the users' needs.
5 Implementation

Once the design had been made and a clustering algorithm had been chosen the next step in the development process was the implementation phase. The final structure for the system that was designed can be seen in Figure 11.

![Diagram of the system structure]

Figure 11: A schematic overview of the system.

5.1 Programming Language and Framework

For the development of the web application Python (Python Software Foundation, 2016) was chosen as a programming language since it is a language for which a lot of packages are available, which makes it very versatile. The microframework Flask (Armin Ronacher, 2016) was chosen as the web-framework to be used.

5.2 Retrieving Results

As described in the ‘Design’ chapter (Chapter 3), the choice was made to use the search results from an existing state-of-the-art search engine. Using these results would allow the target group to access the same information as other search engine users, but in a more accessible manner. While there are many search engines available (Google, Bing, Yahoo, Qwant, Duck Duck Go, and many more) only a few provide original and up-to-date search data. Which ‘Search API’ was chosen will be described in the following subsection.

5.2.1 Choosing a Search API

With the criteria of original and up-to-date search data in mind, search engines can be divided in four categories. The first category contains search engines that provide up-to-date information
Table 3: A comparison in terms of calls, results and costs between the Bing Search API and the Google Custom Search API.

<table>
<thead>
<tr>
<th></th>
<th>Bing</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results returned per call</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>Free calls per month</td>
<td>5000</td>
<td>3100</td>
</tr>
<tr>
<td>Free results per month</td>
<td>250,000</td>
<td>31,000</td>
</tr>
<tr>
<td>Costs for extra calls</td>
<td>€7.65 per 5,000 calls</td>
<td>€22.95 per 3,100 calls</td>
</tr>
<tr>
<td>Costs per extra call</td>
<td>€0.0015 (10 results)</td>
<td>€0.0074 (50 results)</td>
</tr>
<tr>
<td>Costs per 50 results</td>
<td>€0.0370</td>
<td>€0.0074</td>
</tr>
</tbody>
</table>


5.2.2 Calling The Search API

Results are retrieved from the Bing Search API through Representational State Transfer (REST), which is a standard manner for interaction with web resources. An URL is built that contains a request which is made up of, amongst other things, the search query, and that URL is sent to the Bing Search API. The URL that was used for retrieving results for the search interface was constructed as follows:

\[
\text{URL} = \text{Base URL} + \text{Result Type} + \text{Query Start} + \text{Query} + \text{Query End} + \text{Number of Results} + \text{Format} + \text{Market}
\]
The base URL is the location of the Bing Search API. The result type is ‘Web’ since we want web results. Query start and query end surround the query with single apostrophes encoded as ASCII. The number of results indicates how many results should be retrieved and market is set as the Dutch market since the target users are Dutch. Setting the market for the Netherlands does not exclude results in other languages. Setting the market is a similar setting as using Google Netherlands instead of just Google, which means that the results are tuned to what Bing assumes Dutch people would like to see.

Before inserting the search query into the URL it is set to URL-encoding to make sure that it is safe to add. This means that all non-ASCII characters are transformed to ASCII (for example ‘Example :)’ will become ‘Example+%3A%29’). It was also made sure that character that have a special function when searching would still work (for example, the - character that can be used to excluded results that contain a certain term (Google, 2017; Microsoft, 2017) ).

With these parameters, the API returns a dictionary in the JSON format that has the structure that can be seen in Figure 12.

![Figure 12: A schematic representation of the dictionary returned by the Bing Search API.](image)

The search result information needed in the clustering process is then extracted from this dictionary and passed on to the clustering algorithm.

### 5.3 Clustering Algorithm Settings

As described in the previous chapter (Chapter 4), the selected clustering algorithm, Lingo, is part of the Carrot² search results clustering engine. The Carrot² document clustering server (also called Carrot² DCS) can be run on the same server as the search application. Results can then be send to and clusters retrieved from it, again through REST interaction. The URL that is constructed in this communication process can contain many parameters. An overview of these parameters and the values that are set for them (and why these values are chosen) can be found in Appendix C. The choice for the final values of the parameters was based on the preferences for cluster labels that were indicated by participants of the survey (see Appendix A).

### 5.4 Page Structures

In this section the general structure of the web interface pages will be described. These pages were constructed with the guidelines and requirements we have established for usable websites in mind (which were presented in Section 2.5.3 and Chapter 3, respectively). The web interface consists of an index page into which so-called ‘blocks’ for the search page, results page, and error pages can be added. This index page always has a script that lets the page focus on the search field (if present)
and a script that plays a short sound if the page is loaded (a ‘pling’ for loaded correctly or a ‘beep’ for loaded incorrectly).

```html
<div class="block-div">
  <h2>Zoeken</h2>
  <div id="search-box">
    <form action="{{ url_for('index') }}" method="post" aria-label="Zoekveld" role="search">
      {{ form.hidden_tag() }}
      {{ form.search_question(size=40) }}
      <input id="search-input" type="submit" value="Zoek">
    </form>
    <div id="help_and Tips_link">
      <a href="{{ url_for('search_help') }}" id="help_link">
        Klik hier voor uitleg en tips voor het zoeken.
      </a>
    </div>
  </div>
</div>
```

Figure 13: The HTML code for the search area block that is incorporated in the search and results pages.

The search page consists of the index page with the search block added to it. This search block contains the code found in Figure 13. The elements that are encoded are a header (level 2) ‘Zoeken’, a form with one field (which is the search field), and a link to a page with a short text with suggestions for creating a query. A screenshot can be found in Figure 14.

Figure 14: A screenshot of the search page.

To create the results page the search block is added to the index page. Then two checks are performed, namely: Have results been found, and should the ‘first results’ be shown. The first check is to make sure that if there are no results, and therefore also no clusters to be shown, a textual notification is shown that tells the user that no results were found. The loading of a page with this notification is also accompanied by the negative ‘beep’ sound. The second check enables the admin (via the admin panel) to set whether or not the first five results that were returned
should be shown first, before the clusters are displayed.

After these checks are performed and the first results are displayed (or not) the clusters will be displayed. The page will automatically focus on the first results header (if present, otherwise the ‘clustered results’ header) so that the user does not have to search for it. The first results and cluster groups both have a <h2>-tagged heading with respectively ‘Eerste resultaten’ or ‘Gegroepeerde resultaten’. These headings are then followed by the results. For the first results group these are simply an unlabelled list with the first five search results added to that list as items. For the clustered results for each cluster the results are structured as follows: Each cluster name has <h3>-tags, and is followed by an unlabelled list. Each search result in a cluster is then added as a list item to that list. A cluster always ends with a ‘get more search results in this category’-button, which enables the user to search again, but with the cluster name added to the original query (this button is not displayed for the last cluster, which is always called ‘Other’). Two screenshots of the results page can be seen in Figures 15 and 16.

The page for a 404 HTML error also consists of a block (title and message) that is pasted into the index page. The errors for ‘no results found’ and the 503 error returned by the Search API (which indicates that the limit for the number of calls to the Search API has been reached) are included in the results page.

A number of pilots of different versions of the application were held with a domain expert. A number of important suggestions were provided. The first was to add a role-label to the search field, so role="search" was added. Another was that the positive ‘pling’ and negative ‘beep’ sounds were clear in meaning. A third suggestion was to add a language label to the website (‘NL’) to make sure that the screen reader would choose the correct language. The overall conclusion by the expert was that the application seemed to work well, and that the website had less ‘junk’ on it than most websites.

5.5 Admin Panel

The Admin Panel for the search interface was implemented using the flask-admin package. A simple database was constructed to keep track of the username and password needed to access the panel. The database also kept the values for the settings that could be adjusted by the admin in the panel. The database therefore contained tables for:

- Admin login information
- Search results retrieval settings
- Search results clustering settings
- Search interface display settings

Screenshots of the pages in the admin panel can be found in Figures 17-21.

The next step in the development process was to evaluate the web search interface. This will be described in the following chapter.
Figure 15: A screenshot of the beginning of the results page with the ‘first results’ block shown.
Figure 16: A screenshot of the part of the results page following on that shown in Figure 15 containing part of the ‘Grouped results’ section.
Welkom bij het beheerders paneel

U moet ingelogd zijn om deze pagina te kunnen bekijken.

Gebruikersnaam

Wachtwoord

Verzenden

← Terug naar de zoekpagina

Figure 17: A screenshot of the login page in the admin panel before logging in.

Welkom bij het beheerders paneel

Het inloggen is geslaagd.

Gebruik het bovenstaande menu om te navigeren.

← Terug naar de zoekpagina

Figure 18: A screenshot of the login page in the admin panel after successfully logging in.
Zoekpagina Instellingen

Op deze pagina is het mogelijk de instellingen voor de presentatie van de zoekresultaten in te stellen.

Laat de eerste resultaten zien

Ja

Laat related search zien

Nee

Verzenden

Met de volgende knop is het mogelijk de oorspronkelijke waarden weer te herstellen:

Herstel waarden

Figure 19: A screenshot of the page in the admin panel where the settings for the search page itself can be changed.
Bing Instellingen

Op deze pagina is het mogelijk de instellingen voor het ophalen van de Bing resultaten in te stellen.

Aantal opgehaalde zoekresultaten
50

Echte zoekresultaten ophalen
Ja

Er zijn nog zoekvragen over
Ja

Verzenden

Met de volgende knop is het mogelijk de oorspronkelijke waarden voor het ophalen van de zoekresultaten weer te herstellen:

Herstel waardes

Figure 20: A screenshot of the page in the admin panel where the settings for retrieving the Bing search results can be changed.
Carrot$^2$ Instellingen

Op deze pagina is het mogelijk de instellingen van het gebruikte Carrot$^2$ clustering algoritme in te stellen.

Cluster count base
Een factor die gebruikt wordt om het aantal clusters te berekenen op basis van het aantal aangeleverde resultaten. Hoe groter deze waarde des te meer clusters er gemaakt worden. Het aantal clusters is in proportie tot de grote van dit getal, maar niet lineair.

| Waarde: 7 |

Cluster merging threshold
Het percentage aan overlap dat er minimaal tussen twee clusters moet zitten voordat ze samengevoegd worden. Bij een lager percentage zullen de clusters sneller samengevoegd worden.

| Waarde: 0.85 |

Size score sorting ratio
Geeft de balans tussen het sorteren van de clusters op grootte en het sorteren op score weer. Bij een grote waarde is grootte belangrijker, bij een hoge waarde wordt meer waarde gehecht.

Figure 21: A screenshot of the page in the admin panel where the values for the parameters of the Lingo clustering algorithm can be changed.
6 Evaluation

To evaluate the search engine expert reviews were conducted. The web application was reviewed by three experts. These were a Human Media Interaction expert, an expert with experience in designing products for the blind, and a blind expert who advises developers on their products. A review plan was given to these experts that included a short introduction to the task they were given, an example scenario of how an interaction with the application would work, short instructions to explore the application, and a set of questions (the final version of the review form can be found in Appendix D).

The introduction described that the application is a search engine for the blind. It emphasizes that this group of users usually use a screen reader to access the information presented on websites and that the website contains some sound notifications to indicate when it is loaded or when an error occurs.

Following the introduction, the example scenario described in four steps how a user could use the search engine. This started with opening the search page, which focuses on the search field, and hearing a sound when the page is loaded. It was then described how a user would insert their query and perform the search. The scenario ended with a the description of what they would find on the results page, and what their possible actions from there were.

In the exploration instructions the experts were instructed to search for three terms. This made sure that each reviewer would have an amount of basic experience with the web interface. They were told that they could use an operating system, browser and screen reader to their liking. After searching for the three terms they were free to explore the site as they wanted and could perform more searches with their own terms.

6.1 Question Design

The questions for the review were constructed based on existing guidelines for usability and accessibility of websites. Specific guidelines from several papers on designing websites for the blind were also incorporated (for an extensive description see Section 2.5).

The first set of guidelines that were used were the 10 Heuristics of Usability by Nielsen (Preece et al., 2002, p. 27) since they describe the standards to which a usable application should conform. The second set that were used were the Web Content Accessibility Guidelines 2.0 (or WCAG 2.0) since these guidelines are guidelines that were constructed to improve the accessibility of websites (Caldwell et al., 2008). The final group of guidelines that were used were the guidelines from five papers. Three of these papers suggested guidelines for improving search engines specifically for the blind (Leporini et al., 2004, 2008; Gooda Sahib et al., 2012). The other two proposed guidelines that can be used for improving the accessibility of websites for the blind in general (Correani et al., 2004; Theofanos & Redish, 2003).

The process of constructing questions for reviewing consisted of a few steps. The first step was to formulate a corresponding question for each guideline in all of the sets mentioned above. Example questions that could be asked for the 10 Heuristics were also included in this step (Preece et al., 2002, pp. 408-409). The construction of all these questions resulted in 67 potential review questions that contained some overlap (which was to be expected since the guidelines have some common ground).

The second step was to manually cluster the 67 questions by subtopic and remove the duplicate questions. This resulted in seven groups that contained 44 questions. These groups could be characterized by the following descriptions:

- The visibility of the website’s status (4 questions)
• Readability and usability of the content (9 questions)
• The structure of the website (6 questions)
• Usability with a screen reader (8 questions)
• Error messages and error prevention (4 questions)
• Control by and freedom of the user (8 questions)
• Website lay-out and formatting (5 questions)

The final step was then to revise the questions. If for example, three questions contained some overlap and the same information could be gained by combining them into two new questions, this was done. This resulted in four final groups with 28 questions. These groups were the following:

• The visibility of the website’s status (7 questions)
• Readability and usability of the content (8 questions)
• The structure of the website (5 questions)
• Usability with a screen reader (8 questions)

The questions from the three groups that were removed were added to the other groups or recombined with questions from those groups.

The final version of the expert review document was then send to the three expert reviewers and can be viewed in Appendix D.

6.2 Results

The three experts had filled in the expert reviews while using the web interface on their own operating system and using a browser and screen reader of their choice (the advice was given though that, when in doubt as to which screen reader to use, NVDA (NV Access, 2015) was free and had a reasonable learning curve). The questions that they were asked to answer required them to give a grade (on a scale of 1-10) on for the question topic and an explanation if relevant. In the following three subsections the results of the review will be presented. In general, the reviews were positive and most of the questions got a grade of conformance that was either a 7/10 or 8/10 with little comments other than ‘this feature is sufficiently present’ or ‘the application conforms’.

6.2.1 The Good

The application contains no useless content and follows the HTML standards. It also works with all browsers and screen readers. Users can easily locate the search field (which is also due to the ‘focus on feature’). The structure of the website is very clear and consistent. The contents, structure and sounds are functional. The results that are provided for the queries are relevant and the clusters are as well. The first time that a user sees the website the website is - as can be expected - still new to them, but there is not a steep learning curve. After the first view it is clear what can be expected and where items are located on the page.

6.2.2 The Bad

All experts noted that the structure of the page was clear, but that more explanation to the user about the concept might be a good idea. The user has no idea that the website will cluster the results until they encounter these clusters. Another important comment that was made was that the title of the web page should also change for the result page or error pages.
6.2.3 The Suggestions

The website was built to be accessible, but a note was made that it might be too clean and that caution should be taken to avoid the feeling of it being ‘a website for the blind’. The blind prefer a normal website that happens to be accessible/usable to a website that feels like it was necessary to create a simple version for them specifically.

An overview of the cluster headings could prove useful for users that do not have the screen reader or short-key knowledge to create a heading list. Another suggestion that was made was for the addition of more accelerators (for example, short-cuts). One of the reviewers mentioned that users might like to have the option to turn the sounds on or off at will.
7 Discussion and Conclusion

In this thesis the development of a proof of concept search interface for the blind was described. As could be concluded from the 53 respondent survey (see Appendices A and B) that was held during this research, the need for such a web interface is present since it was found that blind and sighted users both performed equal amounts of searches, but that the blind found less often what they were looking for.

To develop a usable proof of concept web search interface that makes searching on the web more accessible for the blind we have taken several steps. We started with an extensive literature research to gain insight into the problems that the blind face when searching online and how these problems might be prevented. This resulted in an explanation of how screen readers work in combination with the web, a realization of the importance of usability as an extension to accessibility, an overview of the problems that arise for blind users both on the web in general and for web search specifically, an insight into the browsing strategies and coping mechanisms that blind users apply when navigating the web, an overview of guidelines that have been suggested in the literature (both for the development of usable websites for the visually impaired in general and search engine specific), and a summary of previous search engines for the blind that have been developed.

The insights that we gained in our literature research were used in making a design for the web interface. We combined the suggestions and guidelines from the literature with the requirements that followed from our definition of the scope, which resulted in a new set of guidelines, and designed the content and structure of the search interface’s pages. Since it followed from our first literature study that a clear structure was essential and that presenting the search results clustered on subtopic might add to that structure, we performed a second literature study with the goal of finding a suitable clustering algorithm. This second literature study resulted in an overview table that showed our requirements and the amount in which the algorithms fulfilled each of them. Based on the fulfilments of these requirements we chose Lingo as the cluster algorithm and described its workings in detail. At the end of Lingo’s clustering process we decided to add an extra step in which we sorted the clusters based on the order of occurrence of the clusters’ first result in the original set of results.

During the implementation phase we applied the guidelines and suggestions to the web search application that we created. We decided to use the Bing Search API to provide us with the search results, added auditory cues that signal when a page has been (un)successfully loaded, and performed a number of pilots with a domain expert. After we had implemented the proof of concept web search interface we performed an expert review with three experts for which we designed a questionnaire. They were asked to test the application using a screen reader and operating system of their own choice. During their test they filled in the questionnaire. The experts’ conclusions were positive. The application contained only useful content, followed standards and guidelines, and worked well with all browsers and screen readers. The structure was very clear and intuitive, although it was noted that caution should be taken to prevent it from being a noticeable ‘application designed for the blind’ since blind users might like it less if it felt like such. A point of criticism was that an explanation of the websites purpose could be added and that the search assistance page could be more extensive. A final suggestion that was made was that maybe an overview could be added with the cluster names to provide a quick overview of the clusters. When relating the review results to the guidelines we listed in Section 2.5.3 and the requirements that we listed in Section 3.3 we can conclude that we have succeeded in the development of a proof-of-concept web search interface that conforms to them.
7.1 Future Research

There are two main directions in which the current research could be extended for future research, which are evaluation and extension.

The first direction (Evaluation) has the highest priority in our opinion. The web search interface has proved through expert reviews to conform to the standards we have set based on our definition of the scope and our literature review, but further evaluation by the target group is necessary. These tests could be performed through large or small scale user testing. Small scale user testing could be performed first by having users test the application using a so-called ‘think out loud’ protocol. Large scale user testing could be performed by letting a large group of participants test the application and to have them fill in a questionnaire. Another useful addition to each of these tests could be to include a comparison with a currently popular search engine (for example, at the time of writing this could be the Google search engine).

The second direction (Extension) focusses on the development of new features for the web search interface. Example directions for extension can be to make the application suited for the partially sighted, to add personal profiles, to add search assistance, or to add other types of search results.

The first of these, extension for the partially sighted, would focus on the addition of features that extend the applications target group to include the partially sighted. It could then be researched how the interface could be adjusted to be clear when used with a magnifying glass or enhanced contrast settings. For a start, this would mean that the best fonts, font-sizes, spacing, and colours should be found to work with those tools.

The second type of extension could be to add personal profiles. A user could set their preferences in their profile and in this manner fine-tune the website to these personal preferences. They could, for example, adjust the volume of the positive and negative feedback sounds, the language settings for their search (for example, show only Dutch results), and in case of a partially sighted user the features suggested above. A user could also be allowed to select if they would like less or more clusters, longer or shorter cluster names, if they would like to see an overview of clusters names, etcetera. Of course, a simple version of these profiles could be created by letting users choose between pre-designed profiles (for example, one for the blind and one for the partially sighted).

The third addition could be to provide more extensive search assistance. This could be done by providing information on how to formulate queries (for example, how to use special characters such as ‘-’ to leave out results with a certain term). What could also be added is query expansion, which is a technique by which queries are automatically extended to retrieve a more complete set of results, for example, by also searching for synonyms of terms.

The fourth extension could focus on making other types of search results than just the textual links accessible. An example could be to improve the accessibility of image or video search. This does provide us with some technical challenges since a description of these media would be needed in order to present the user. Such a description would have to be either added by hand or automatically generated, which are both challenging tasks.

A final line of extension research could be to research if the clustering algorithm itself could be improved. While we choose the best algorithm based on information provided in the literature, a comparison between the performance of various existing clustering algorithms (plugged into the developed interface) could provided insight on how an improvement might be achieved.

For now, we can conclude that the development of a designated web search interface is an interesting area of research and as the suggestions mentioned above show there are more than enough directions in which research can be continued so that eventually the web can genuinely be seen as an unlimited source of information which is equally accessible for the sighted and the visually impaired.
References


Cobos, C., Andrade, J., Constat, W., Mendoza, M., & León, E. (2010). Web document clustering based on global-best harmony search, K-means, frequent term sets and Bayesian information criterion. 2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010


Di Marco, A., & Navigli, R. (2013). Clustering and Diversifying Web Search Results with Graph-Based Word Sense Induction. doi: 10.1162/COLI


Duck Duck Go. (2016). Duck duck go. [https://duckduckgo.com/](https://duckduckgo.com/). (Online; accessed online 02-08-2016)


Qwant. (2016). *Qwant.* [https://www.qwant.com/](https://www.qwant.com/). (Online; accessed online 02-08-2016)

Rømen, D., & Svanæs, D. (2012). Validating WCAG versions 1.0 and 2.0 through usability


A The Survey

A.1 Aim

A survey was conducted in the Netherlands among blind, partially sighted and sighted users of the internet with the aim to gather information on two main questions:

- How do the respondents search and what do they experience?
- What are the respondents’ preferences for group names? (These are the group names that follow from the clustering of search results applied in the developed application.)

With the help of Dedicon’s external communication department the target group was approached.

In the sections below we will first describe the design of the survey and the formulation of the questions. Next, we present and discuss the results.

A.1.1 Design and Question Formulation

One of the first decisions that was made for this survey was to set a limit of 20 questions, since having more questions would increase the chance of respondents quitting halfway through.

The final version of the survey consisted of five pages, which contained 20 questions in total. This final version can be found in Appendix B. The set-up for the pages in the survey was as follows:

- Page 1: Introduction (no questions)
- Page 2: Background information (two or three questions, depending on level of visual impairment)
- Page 3: Questions about search on the internet (five questions)
- Page 4: Questions about the group names (seven questions)
- Page 5: Questions about preferences (five questions)

In what follows we will explain how the survey questions were constructed.

Questions About Online Search

The questions on Page 2 and 3 of the survey aimed to gather information in order to answer the ‘How do the respondents search and what do they experience?’ question. Page 2 contained three questions that could be used to differentiate the blind respondents from the sighted. An answering option for the partially sighted was added as well. When respondents would either pick the blind or partially sighted option they would also be shown a question that asked them about the computer aids they used (screen reader, refreshable braille display, magnifying software, enhanced contrast settings, none). A question was also added on this page about age since this might have an influence on the respondents’ computer skills.

The questions on Page 3 were about how users searched on the internet and if they encountered problems while doing so. Respondents were asked how often they searched on the internet - as a measure of experience. They were also asked how much they agreed with the statement that they could always find, in one attempt, what they were looking for. The distinction made in these two questions was important, since respondents might have a lot of experience with searching, but they could still feel like it is hard to find what they are looking for. The paper by Buzzi et al. (2004) states that the blind in that study tended to find less often what they were looking for than the

---

4A foundation focussed on making information accessible for people with a reading disability (www.dedicon.nl).
sighted. 92% of their sighted respondents answered ‘yes’ on the statement ‘Search engines are easy to use’, while 76% of the blind respondents stated ‘not always’. Furthermore, 90% of the sighted respondents answered ‘almost always’ to the question if they could find the information they were looking for, but for the blind respondents 38% answered ‘sometimes’ and 8% answered ‘almost never’. Something similar might also be the case for the blind in the Netherlands.

The other three questions on Page 3 aimed to find out which search engine respondents used, what made them chose that search engine, and what difficulties they experienced when searching.

Questions About Preferences
The goal for the questions on Page 4 and 5 was to find out the preferences of the respondents for the group names that would result from the clustering of the search results.

On Page 4 these preferences were questioned indirectly by three types of questions. In the first type of question respondents were asked to indicate which set of group names they would prefer most for a certain query. The second type of question was an open question, which asked them why they made certain choices and what they found important aspects. The third type of question gave a list of possible group names and asked respondents to choose four of them.

Initially the plan was to enquire about two settings for four parameters (Cluster Count Base, Phrase Label Boost, Title Word Boost, Maximum Word Document Frequency) using questions. This would result in 16 questions that would have four sets of group names. Respondents would then be asked to choose the one they preferred. Since 16 questions would - when combined with the background and search questions - exceed the limit of 20 questions, it was decided to vary the parameters in groups of two. This meant that eight questions would be enough. The parameters that influenced the number of clusters and the length of the names were grouped. The other grouping was that of the parameters that influenced the use of words from the search result titles and the use of often occurring words from in the search results. That is, one size and length grouping and one content grouping. The scheme used for formulating the choice options per question (and thus the underlying parameter combinations) can be found in Table 4.

When the version of the survey with the eight ‘four options’-questions was discussed with an expert on accessible surveys, the conclusion was that the type of question with four choice options was already difficult for the sighted, but would be nearly impossible to handle for the blind. After the conclusion that the questions would be too difficult to fill in when using a screen reader, the questions were redesigned. It was decided to remove one parameter from the set that was going to be tested (the one influencing the number of labels, since it had initially been added to gather extra information, but was not considered to be essential). Another decision that was made was to limit the amount of choice options to two per question. The questions would thus include two options and the respondents would be asked to indicate their preference on the following scale:

- I really prefer Option 1
- I somewhat prefer Option 1
- I do not have a preference
- I somewhat prefer Option 2
- I really prefer Option 2

One reason to add the scale was to prevent that respondents would just randomly choose one option. The other reason was that this gave information about the degree of preference, which might prove informative. The changes resulted in a new scheme for four questions to evaluate the first of the three parameters left (Phrase Label Boost, see Table 5). A question was also added to find out the motivation for the respondents’ choices.
Since the choice options that resulted from the variation of the other two parameters (Title Word Boost and Maximum Word Count) were very similar and only varied in one term per option, the choice was made to make a single question of the ‘choose four’-type. For this question another question about the motivation was added.

On Page 5 the preferences were questioned directly using two types of questions. In the first type respondents were given a statement about an influence that the clustering settings could have on the group names (specificity, overlap, number of words, number of groups). They had to indicate how much they agreed with the statements. The second type of question was a question that asked them to indicate which influences, from a list of possible influences, they found most important.

A.2 Results

A.2.1 Respondents

The survey was filled in by 53 respondents. Of these respondents 20 indicated that they were blind, 15 indicated that they were partially sighted, and 18 answered that they were sighted (Level of Sightedness (Blind, Partially Sighted, Sighted)). The age distribution (Age) within these groups can be seen in Figure 22. As can be seen, the distribution within the blind and partially sighted groups is relatively similar. The mode for the sighted group lies slightly higher.

A.2.2 Statistical Tests Used

The questions in the survey that were analysed using a statistical test were all Likert-scale questions. This meant that the dependent variable was ordinal. In all cases, the independent variable was Level of Sightedness, with three options (blind, partially sighted, sighted). Independence of observations was guaranteed and the age distribution within the groups was not homogeneous. Therefore a Kruskal-Wallis H test comparing modes was used.

A.2.3 Results - Online Search

The first question about online search asked respondents how often they searched online (see Figure 23 for a histogram).

The Kruskal-Wallis H test showed that there was no statistically significant difference in the number of searches per day between the different groups, $\chi^2(2) = 1.329, p = 0.515$, with a mean rank amount of searches per day of 24.15 for the Blind, 27.47 for the Partially Sighted and 29.78 for the Sighted.

The second question asked how much respondents agreed with the statement ‘I can always find what I am searching for in one attempt, when searching online.’ Figure 24 shows the histogram corresponding to the answers for each of the groups to this question.

The Kruskal-Wallis H test showed that there was no statistically significant difference in the amount of agreement between the different groups, $\chi^2(2) = 5.292, p = 0.071$, with a mean rank agreement score of 32.00 for the Blind, 27.40 for the Partially Sighted and 21.11 for the Sighted.

Since the significance score was low ($p = 0.071$) and this test compared all three groups another comparison was performed between the Blind and Sighted groups. This Kruskal-Wallis H test showed that there was a statistically significant difference in the amount of agreement between the two groups, $\chi^2(2) = 4.109, p = 0.043$, with a mean rank agreement score of 22.73 for the Blind, and 15.92 for the Sighted.
On the question: ‘Which search engine do you use?’ all but two of the respondents answered ‘Google’. One respondent (partially sighted) stated that he/she used ‘Startpage’ and one (blind) answered ‘Duck Duck Go’ (which combines search results from Bing and Google).

On the (optional) open question: ‘Are there things in search engines that sometimes make searching difficult for you?’ the responses were coherent within each of the three groups. The Blind named annoyances such as: Results not shown first, commercial links, pop-ups, dynamic suggestions do not work with my screen reader, studying the search results can be difficult. The Partially Sighted named: commercial links, no overview, and many results. The Sighted named fewer annoyances and these could mostly be categorized as ‘results are not relevant’, and ‘commercial links’.

Figure 22: Histograms of the respondents’ age for each of the three respondent groups (blind, partially sighted and sighted).
A.2.4 Results - Preferences

The first four questions in this category asked the respondents to indicate their preference for either ‘option 1’ or ‘option 2’. Each of these options had a group of possible cluster labels that could be generated for a query (that was also listed in the question). The participant was shown the query and the two options and could chose on a scale, ranging from ‘I strongly prefer Option 1’ to ‘I strongly prefer Option 2’. The four questions were shown in a random order. The four queries were the following: ‘Trein naar Londen’ (Query 1), ‘Koopzondagen Utrecht’ (Query 2), ‘Recept pasta met zalm’ (Query 3), ‘Wat kost een paspoort’ (Query 4). The distributions for preferences per question can be observed in Figures 25-28. As can be seen, there are differences in preferences between the groups. Two sets of Kruskal-Wallis H tests were run. The first set tested for a statistically significant difference in amount of agreement between all three of the groups. The second set compared just the Blind and Sighted groups. The results for these tests can be found in
Tables 6 and 7. As can be seen, the Kruskal-Walls H tests with all three groups included showed a statistically significant difference in the amount of agreement in the ‘Koopzondagen Utrecht’ case. The second set of tests shows that there is a statistically significant difference in the agreement for the first three cases.

On the open question ‘What did you find important when indicating your preference for Option 1 or 2?’ the groups were in agreement. The cluster names had to be relevant (having the search term repeated in the cluster name was a plus). The names had to be relatively specific, but also not too long.

The next question asked participants to choose four cluster labels out of nine. The distribution of those choices can be found in Figure 29. As can be seen, the participants tend to agree on which labels should be in the top 5.

When asked why they had chosen those four labels the participants more or less agreed that the labels had to be relevant, had to suggest that the groups contained ‘quality’ sites (not too many commercials or with a bad layout) that could provide them with prices and specifications. They wanted to gather various types of information about the product and wanted objective information.

The last questions were specifically about the features of the clusters (specific names, names overlapping in meaning allowed, labels with multiple words, few clusters). For each of these features a statement was formulated, for example, ‘I would like to see specific cluster names’. The participants were then asked to indicate on a 5-point Likert-scale how much they agreed with the statement. The percentages for each of these three groups for all of the features are displayed in Table 8. As can be seen in Figure 30, where the ‘Strongly’ and ‘Mildly’ answers have been grouped together, the three groups of respondents are in agreement with each other. They agree that they would like to see specific cluster labels, that they would not mind some overlap in the meaning of cluster labels, they are positive to neutral about the length of the labels, and they would like to see few clusters.

The last question asked participants to select two cluster features (out of six) that they found the most important. Most participants (in all groups) indicated that they found the relevance and specificity very important.

A.3 Conclusions

There are a few conclusions that can be drawn from the survey. These will be explained in the following two subsections. In Section A.3.1 the conclusions for the part of the survey about online search will be presented. In Section A.3.2 the results for the questions about the cluster preferences will be presented.

A.3.1 Conclusions - Online Search

There is no significant difference in the number of searches that the blind, partially sighted and sighted perform per day. There is, however a significant difference in the level of agreement on the statement ‘I can always find what I am searching for in one attempt, when searching online’ between the blind and the sighted. The mode for the answers by the blind lies at the neutral answer, while the mode for the sighted lies at the ‘agree’ category. So, while both groups on average perform the same number of online searches per day, the blind less often find what they are looking for in one attempt. When considering the skewed difference in ages between the groups this is even more remarkable because the sighted group is older, and you might expect that if there was any influence of age, older people would find less often what they were searching for than younger people. Most respondents use Google and the annoyances of the blind seem more inhibiting for
A.3.2 Conclusions - Preferences

These conclusions will be presented in two parts. The first part will focus on the differences between the blind and sighted in their answers on questions about the queries with two different options for a set of cluster labels. The second step will focus on the question in which the participants had to choose four of nine possible labels and on the questions in which the participants had to indicate the level of agreement on the four statements about cluster features.

The blind and sighted disagreed on three of the four questions in which they had to choose between an ‘Option 1’ set of cluster labels and an ‘Option 2’ set of labels. In the case of ‘Trein naar Londen’ the blind preferred the option that contained ‘Eurostar naar Londen’ and ‘Treinreis Londen’, while the sighted preferred the option that included ‘Comfortabel met de Trein’. This difference could be due to the impression of the blind that ‘Comfortabel met de Trein’ could just contain commercial links instead of more straight to the point information about trains that go to London.

In the case of ‘Koopzondagen Utrecht’ the blind preferred ‘Overzicht van alle Koopzondagen’ en ‘Bekijk de Openingstijden’, while the sighted preferred the Option that contained ‘Juni 2016’ and ‘Koopzondagen Utrecht 2016’. The difference in this case might also be because the labels preferred by the blind sound like they contain a display of information (a list of dates, and a list of times), while the labels preferred by the sighted could give you pages that inform you that the shops are opened in Utrecht on some Sundays, but without the actual dates.

In the case of ‘Recept Pasta met Zalm’ the blind and sighted groups both preferred the set of labels that contained ‘Pasta met Gerookte Zalm’ instead of ‘Gerookte Zalm’, but the blind also chose the ‘I prefer both options equally’ option a lot. The conclusion that can be drawn from this is that both groups preferred the longer version of the label that included more of the original query.

In the case of ‘Wat Kost een Paspoort’ the blind and sighted again both preferred the set of labels that contained ‘Paspoort en Indentiteitskaart’ and ‘Nieuw Paspoort’ instead of ‘Paspoort en ID-kaart’ and ‘Producten en Diensten’. Again the sighted were more outspoken in their preference than the blind who again leaned more towards the neutral preference. The favour for the first option is probably due to a preference for ‘Nieuw Paspoort’ over ‘Producten en Diensten’.

The ‘chose four out of nine’ question and the questions in which the participants had to indicate how much they agreed with the four given statements resulted in the same conclusions. Participants indicated that they preferred specific cluster labels, would not mind some overlap, would not mind long labels (but they should not be too long) and they did not want too many clusters.
<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
<th>Parameter 1</th>
<th>Parameter 2</th>
<th>Parameter 3</th>
<th>Parameter 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>1a</td>
<td>2a</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>1a</td>
<td>2a</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>1.3</td>
<td>1a</td>
<td>2a</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>1a</td>
<td>2a</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
<td>1a</td>
<td>2b</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>1a</td>
<td>2b</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>2.3</td>
<td>1a</td>
<td>2b</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>2.4</td>
<td>1a</td>
<td>2b</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td>3</td>
<td>3.1</td>
<td>1b</td>
<td>2a</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td>1b</td>
<td>2a</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>3.3</td>
<td>1b</td>
<td>2a</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>3.4</td>
<td>1b</td>
<td>2a</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td>4</td>
<td>4.1</td>
<td>1b</td>
<td>2b</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>4.2</td>
<td>1b</td>
<td>2b</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>4.3</td>
<td>1b</td>
<td>2b</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>4.4</td>
<td>1b</td>
<td>2b</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td>5</td>
<td>5.1</td>
<td>1a</td>
<td>2a</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>5.2</td>
<td>1a</td>
<td>2b</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>5.3</td>
<td>1b</td>
<td>2a</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>5.4</td>
<td>1b</td>
<td>2b</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td>6</td>
<td>6.1</td>
<td>1a</td>
<td>2a</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>6.2</td>
<td>1a</td>
<td>2b</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>6.3</td>
<td>1b</td>
<td>2a</td>
<td>3a</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>6.4</td>
<td>1b</td>
<td>2b</td>
<td>3a</td>
<td>4a</td>
</tr>
<tr>
<td>7</td>
<td>7.1</td>
<td>1a</td>
<td>2a</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>7.2</td>
<td>1a</td>
<td>2b</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>7.3</td>
<td>1b</td>
<td>2a</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td></td>
<td>7.4</td>
<td>1b</td>
<td>2b</td>
<td>3b</td>
<td>4a</td>
</tr>
<tr>
<td>8</td>
<td>8.1</td>
<td>1a</td>
<td>2a</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>8.2</td>
<td>1a</td>
<td>2b</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>8.3</td>
<td>1b</td>
<td>2a</td>
<td>3b</td>
<td>4b</td>
</tr>
<tr>
<td></td>
<td>8.4</td>
<td>1b</td>
<td>2b</td>
<td>3b</td>
<td>4b</td>
</tr>
</tbody>
</table>

Table 4: The initial eight questions and the parameter combinations that were used per choice option.
Table 5: The four questions in the renewed version and the parameter combinations that were used per choice option.

<table>
<thead>
<tr>
<th>Question</th>
<th>Answer</th>
<th>Parameter 1</th>
<th>Parameter 2</th>
<th>Parameter 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.1</td>
<td>1a</td>
<td>2a</td>
<td>3a</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>1b</td>
<td>2a</td>
<td>3a</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
<td>1a</td>
<td>2a</td>
<td>3b</td>
</tr>
<tr>
<td></td>
<td>2.2</td>
<td>1b</td>
<td>2a</td>
<td>3b</td>
</tr>
<tr>
<td>3</td>
<td>3.1</td>
<td>1a</td>
<td>2b</td>
<td>3a</td>
</tr>
<tr>
<td></td>
<td>3.2</td>
<td>1b</td>
<td>2b</td>
<td>3a</td>
</tr>
<tr>
<td>4</td>
<td>4.1</td>
<td>1a</td>
<td>2b</td>
<td>3b</td>
</tr>
<tr>
<td></td>
<td>4.2</td>
<td>1b</td>
<td>2b</td>
<td>3b</td>
</tr>
</tbody>
</table>

Figure 24: Histograms of the level of agreement on the statement ‘I can always find what I am searching for in one attempt, when searching online’ by the respondents for each of the three respondent groups (blind, partially sighted and sighted).
<table>
<thead>
<tr>
<th>Query</th>
<th>Test Statistics</th>
<th>Mean Rank Agreement <em>(Blind, P. Sighted, Sighted)</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>$\chi^2(2) = 5.860$, $p = 0.53$</td>
<td>(21.35, 27.50, 32.86)</td>
</tr>
<tr>
<td>Query 2</td>
<td>$\chi^2(2) = 7.197$, $p = 0.027$</td>
<td>(32.10, 29.27, 19.44)</td>
</tr>
<tr>
<td>Query 3</td>
<td>$\chi^2(2) = 3.833$, $p = 0.147$</td>
<td>(22.05, 28.53, 31.22)</td>
</tr>
<tr>
<td>Query 4</td>
<td>$\chi^2(2) = 3.568$, $p = 0.168$</td>
<td>(28.08, 31.57, 22.00)</td>
</tr>
</tbody>
</table>

Table 6: The results for the Kruskal-Walls H test with all three groups included for each of the four ‘Indicate a preference for Option 1 or Option 2’ questions.

<table>
<thead>
<tr>
<th>Query</th>
<th>Test Statistics</th>
<th>Mean Rank Agreement <em>(Blind, Sighted)</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>$\chi^2(1) = 6.346$, $p = 0.012$</td>
<td>(15.50, 23.94)</td>
</tr>
<tr>
<td>Query 2</td>
<td>$\chi^2(1) = 6.987$, $p = 0.008$</td>
<td>(23.90, 14.61)</td>
</tr>
<tr>
<td>Query 3</td>
<td>$\chi^2(1) = 4.094$, $p = 0.043$</td>
<td>(16.20, 23.17)</td>
</tr>
<tr>
<td>Query 4</td>
<td>$\chi^2(1) = 1.681$, $p = 0.195$</td>
<td>(21.63, 17.14)</td>
</tr>
</tbody>
</table>

Table 7: The results for the Kruskal-Walls H test with the *Blind* and *Sighted* groups included for each of the four ‘Indicate a preference for Option 1 or Option 2’ questions.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Group</th>
<th>Strongly Agree</th>
<th>Mildly Agree</th>
<th>Neutral</th>
<th>Mildly Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific</td>
<td>Blind</td>
<td>40%</td>
<td>35%</td>
<td>20%</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>P. Sighted</td>
<td>33%</td>
<td>40%</td>
<td>27%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>Sighted</td>
<td>39%</td>
<td>56%</td>
<td>0%</td>
<td>0%</td>
<td>6%</td>
</tr>
<tr>
<td>Overlap</td>
<td>Blind</td>
<td>25%</td>
<td>45%</td>
<td>15%</td>
<td>10%</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>P. Sighted</td>
<td>7%</td>
<td>53%</td>
<td>27%</td>
<td>0%</td>
<td>13%</td>
</tr>
<tr>
<td></td>
<td>Sighted</td>
<td>0%</td>
<td>61%</td>
<td>17%</td>
<td>17%</td>
<td>6%</td>
</tr>
<tr>
<td>Multi-word</td>
<td>Blind</td>
<td>15%</td>
<td>25%</td>
<td>55%</td>
<td>5%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>P. Sighted</td>
<td>7%</td>
<td>40%</td>
<td>53%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>Sighted</td>
<td>11%</td>
<td>56%</td>
<td>28%</td>
<td>0%</td>
<td>6%</td>
</tr>
<tr>
<td>Few clusters</td>
<td>Blind</td>
<td>15%</td>
<td>25%</td>
<td>25%</td>
<td>30%</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td>P. Sighted</td>
<td>13%</td>
<td>33%</td>
<td>33%</td>
<td>20%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>Sighted</td>
<td>11%</td>
<td>61%</td>
<td>28%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

Table 8: The percentages of agreement for each group with the four statements on cluster features.
Figure 25: Histograms for Option that participants preferred for the ‘Trein naar Londen’ query. The preference scale ranges from 1+++ (I strongly prefer Option 1) to 2+++ (I strongly prefer Option 2).
Figure 26: Histograms for Option that participants preferred for the ‘Koopzondagen Utrecht’ query. The preference scale ranges from 1+++ (I strongly prefer Option 1) to 2+++ (I strongly prefer Option 2).
Figure 27: Histograms for Option that participants preferred for the ‘Recept pasta met zalm’ query. The preference scale ranges from 1+++ (I strongly prefer Option 1) to 2+++ (I strongly prefer Option 2).
Figure 28: Histograms for the Option that participants preferred for the ‘Wat kost een paspoort’ query. The preference scale ranges from 1+++ (I strongly prefer Option 1) to 2+++ (I strongly prefer Option 2).
Figure 29: A histogram for the four out of nine cluster labels that participants chose for the ‘Laptop Kopen’ query.
Figure 30: Four graphs that show the percentage of agreement for each of the three groups of respondents for each of the four statements on cluster features.
B Survey Questions

Pagina 1: Introductie
Welkom!

Deze enquête gaat over hoe mensen zoeken op internet. Graag zouden we van jou weten hoe je zoekt en wat jouw voorkeuren zijn voor de naamgeving van eventuele groepen waarin zoekresultaten ingedeeld zouden kunnen worden.

Pagina 2: Achtergrondinformatie
Eerst willen we je graag een paar achtergrondvragen stellen.

Vraag 1: Wat is je leeftijd?
- Jonger dan 18 jaar
- 19 tot 24 jaar
- 25 tot 34 jaar
- 35 tot 44 jaar
- 45 tot 54 jaar
- 55 tot 64 jaar
- 65 jaar of ouder

Vraag 2: Heb je een visuele beperking?
- Ja, ik ben blind
- Ja, ik ben slechtziend
- Nee

[Als er bij vraag 2 een van de ‘Ja’-antwoorden gekozen is]

Vraag 3: Maak je gebruik van assistentiesoftware zoals een schermlezer, een brailleleesregel, vergrotingssoftware of instellingen voor verhoogd contrast? (Je kunt meerdere opties aanvinken.)
- Ja, ik maak gebruik van een schermlezer
- Ja, ik maak gebruik van een brailleleesregel
- Ja, ik maak gebruik van vergrotingssoftware
- Ja, ik maak gebruik van instellingen voor verhoogd contrast
- Nee, ik gebruik de genoemde assistentiesoftware niet

Pagina 3: Vragen over het zoeken op internet
Nu zouden we je graag vijf vragen stellen over het zoeken op internet.

Vraag 4: Hoe vaak zoek je gemiddeld op internet?
- Minder dan 1 keer per dag
- 1 tot 2 keer per dag
- 3 tot 5 keer per dag
- 6 tot 10 keer per dag
- 11 tot 20 keer per dag
- Meer dan 20 keer per dag
Vraag 5: Geef hieronder aan in hoeverre je het eens bent met de volgende stelling: ‘Ik kan bij het zoeken op internet altijd in één keer vinden wat ik zoek.’

- Heel erg eens
- Redelijk mee eens
- Neutraal
- Redelijk mee oneens
- Heel erg mee oneens

Vraag 6: Welke zoekmachine gebruik je meestal?

- Google
- Bing
- Yahoo
- Duck Duck Go
- Qwant
- Ilse
- Anders, namelijk...

Vraag 7: Waarom kies je meestal voor die zoekmachine?

...

Vraag 8: Zijn er eigenschappen die het zoeken voor je soms lastig maken?

...

Pagina 4: Vragen over groepsnamen

Nu volgen wat vragen over je voorkeuren voor de namen van groepen met zoekresultaten. De groepen worden gemaakt binnen de zoekresultaten, dus de groepsnamen zullen hierdoor ook binnen het onderwerp van de zoekvraag vallen.


Je krijgt eerst vier vragen waarin we je vragen om aan te geven welk van de twee lijstjes met groepsnamen je voorkeur heeft. Daarna krijg je een vraag waarin je bij een zoekvraag de beste vier uit negen gegeven groepsnamen moet kiezen.

Note: De volgende vier vragen worden steeds in een random volgorde aangeboden.

Vraag 9: Er is gezocht op ‘trein naar Londen’. Welk van de volgende twee lijstjes met groepsnamen heeft je voorkeur?

1. Eurostar, comfortabel met de trein, treintickets naar Londen, stedentrip Londen, overige resultaten.
2. Eurostar naar Londen, treinreis Londen, treintickets naar Londen, stedentrip Londen, overige resultaten.

- Optie 1 heeft heel erg mijn voorkeur
- Optie 1 heeft redelijk mijn voorkeur
Ik heb een even grote voorkeur voor optie 1 en optie 2
Optie 2 heeft redelijk mijn voorkeur
Optie 2 heeft heel erg mijn voorkeur

Vraag 10: Er is gezocht op ‘koopzondagen Utrecht’. Welk van de volgende twee lijstjes met groepsnamen heeft je voorkeur?

2. Overzicht van alle koopzondagen, Utrecht 2016, bekijk de openingstijden, zondag van de maand, overige resultaten.

Optie 1 heeft heel erg mijn voorkeur
Optie 1 heeft redelijk mijn voorkeur
Ik heb een even grote voorkeur voor optie 1 en optie 2
Optie 2 heeft redelijk mijn voorkeur
Optie 2 heeft heel erg mijn voorkeur

Vraag 11: Er is gezocht op ‘recept pasta met zalm’. Welk van de volgende twee lijstjes met groepsnamen heeft je voorkeur?

1. Gerookte zalm, pasta met zalm en spinazie, tagliatelle met zalm, kook de pasta beetgaar, overige resultaten.
2. Pasta met gerookte zalm, pasta met zalm en spinazie, tagliatelle met zalm, kook de pasta beetgaar, overige resultaten.

Optie 1 heeft heel erg mijn voorkeur
Optie 1 heeft redelijk mijn voorkeur
Ik heb een even grote voorkeur voor optie 1 en optie 2
Optie 2 heeft redelijk mijn voorkeur
Optie 2 heeft heel erg mijn voorkeur

Vraag 12: Er is gezocht op ‘wat kost een paspoort’. Welk van de volgende twee lijstjes met groepsnamen heeft je voorkeur?

1. Paspoort en identiteitskaart, aanvragen en ophalen, digitaal loket, nieuw paspoort, overige resultaten.
2. Paspoort of ID-kaart, aanvragen en ophalen, digitaal loket, producten en diensten, overige resultaten.

Optie 1 heeft heel erg mijn voorkeur
Optie 1 heeft redelijk mijn voorkeur
Ik heb een even grote voorkeur voor optie 1 en optie 2
Optie 2 heeft redelijk mijn voorkeur
Optie 2 heeft heel erg mijn voorkeur

Vraag 13: Waar lette je op bij het maken van je keuze tussen de verschillende lijstjes met groepsnamen?

...

Vraag 14: Hieronder staan negen mogelijke groepsnamen bij de zoekvraag ‘laptop kopen’. Vink de vier groepsnamen aan die je het liefst zou zien.
Vraag 15: Waarom heb je bij de vorige vraag voor die vier groepsnamen gekozen?

... 

**Pagina 5: Vragen over voorkeuren**

We zouden nu graag van je willen weten in hoeverre je het eens bent met de volgende vier stellingen en welke eigenschappen je het belangrijkst vindt voor de groepsnamen.

Vraag 16: Geef hieronder aan in hoeverre je het eens bent met de volgende stelling:
‘Ik zou graag groepsnamen zien die erg specifiek zijn.’

• Heel erg mee eens
• Redelijk mee eens
• Neutraal
• Redelijk mee oneens
• Heel erg mee oneens

Vraag 17: Geef hieronder aan in hoeverre je het eens bent met de volgende stelling:
‘Ik vind het niet erg als er overlap zit in de betekenis van de groepsnamen.’

• Heel erg mee eens
• Redelijk mee eens
• Neutraal
• Redelijk mee oneens
• Heel erg mee oneens

Vraag 18: Geef hieronder aan in hoeverre je het eens bent met de volgende stelling:
‘Ik zou graag groepsnamen willen zien die uit meerdere woorden bestaan.’

• Heel erg mee eens
• Redelijk mee eens
• Neutraal
• Redelijk mee oneens
• Heel erg mee oneens

Vraag 19: Geef hieronder aan in hoeverre je het eens bent met de volgende stelling:
‘Ik zou graag weinig groepen zien.’

• Heel erg mee eens
• Redelijk mee eens
• Neutraal
• Redelijk mee oneens
• Heel erg mee oneens
• Heel erg mee oneens

Vraag 20: Kies de twee eigenschappen die je het belangrijkst vindt voor de lijstjes met groepsnamen:

• Het aantal namen
• Hoe relevant de namen zijn bij de zoekvraag
• Hoe specifiek de namen zijn
• Hoeveel overlap er tussen de namen is
• De lengte van de namen
• Anders, namelijk...

Pagina 6: Bedankt voor het invullen!
Bedankt voor het invullen van de enquête!
C Lingo Parameter Settings

There are many (29) parameters that can be set for the Lingo algorithm. Since there was no standard list the parameters (and their ranges/types) were retrieved from the workbench version of Carrot$^2$. The final settings for the parameters can be found in Table 9$^5$.

Most of the parameters were kept at their standard values, but a few were tweaked. These were:

- Cluster Count Base: The standard setting provided many clusters. To limit this this value was set to a value that made it output about 5 to 7 clusters.
- Cluster Merging Threshold (was 0.7, is 0.85): Clusters meanings were allowed to overlap a little and the coherence within the clusters improved by using a slightly higher value.
- Phrase Label Boost (was was 1.5, is 3.0). Labels were just one word even if two words would have been better.
- Phrase Document Frequency Threshold (was 1, is 3). To improve the specificity.
- Minimum Cluster Size (was 2, is 3). Clusters should contain at least three results.
- Word Document Frequency Threshold (was 1, is 3). To exclude, for example, very often occurring verbs.

$^5$For two parameters the names of the various options were abbreviated since they were very long. The abbreviations stand for the following: NMFEDF: NonnegativeMatrixFactorizationEDFactory, PSVDF: PartialSingularValueDecompositionFactory, NMFKL: NonnegativeMatrixFactorizationKLFactory, LNMFF: LocalNonnegativeMatrixFactorizationFactory, and KMMFF: KMeansMatrixFactorizationFactory. The same was done for the Term Weighting parameter. LogTITW: LogTfIdfTermWeighting, LinTITW: LinearTfIdfTermWeighting, and ITW: TfTermWeighting.
<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Type</th>
<th>Range</th>
<th>Initial Value</th>
<th>Final Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Count Base</td>
<td>Integer</td>
<td>2-100</td>
<td>30</td>
<td>7</td>
</tr>
<tr>
<td>Cluster Merging Threshold</td>
<td>Decimal</td>
<td>0.0-1.0</td>
<td>0.7</td>
<td>0.85</td>
</tr>
<tr>
<td>Size-Score Sorting Ratio</td>
<td>Decimal</td>
<td>0.0-1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Cluster Label Assignment Method</td>
<td>String</td>
<td>UniqueLabel</td>
<td>UniqueLabel</td>
<td>UniqueLabel</td>
</tr>
<tr>
<td>Phrase Label Boost</td>
<td>Decimal</td>
<td>0.0-10.0</td>
<td>1.5</td>
<td>3.0</td>
</tr>
<tr>
<td>Phrase Length Penalty Start</td>
<td>Integer</td>
<td>2-8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Phrase Length Penalty Stop</td>
<td>Integer</td>
<td>2-8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Remove Labels Ending In Genetive Form</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Remove Leading And Trailing Stop Words</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Remove Numeric Labels</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Remove Query Words</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Remove Short Labels</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Remove Stop Labels</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Remove Truncated Phrases</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Title Word Boost</td>
<td>Decimal</td>
<td>0.0-10.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Truncated Label Threshold</td>
<td>Decimal</td>
<td>0.0-1.0</td>
<td>0.65</td>
<td>0.65</td>
</tr>
<tr>
<td>Factorization Method</td>
<td>String</td>
<td>NMFEDF \lor PSVDF \lor NMFKLFP \lor LNMFF \lor KMMFF</td>
<td>NMFEDF</td>
<td>NMFEDF</td>
</tr>
<tr>
<td>Factorization Quality</td>
<td>String</td>
<td>Low-High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Maximum Matrix Size</td>
<td>Integer</td>
<td>5000-2147483647</td>
<td>37500</td>
<td>37500</td>
</tr>
<tr>
<td>Maximum Word Document Frequency</td>
<td>Decimal</td>
<td>0.0-1.0</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Term Weighting</td>
<td>String</td>
<td>LogTITW \lor LinTITW \lor ITW</td>
<td>LogTITW</td>
<td>LogTITW</td>
</tr>
<tr>
<td>Default Clustering Language</td>
<td>String</td>
<td>English</td>
<td>English</td>
<td>English</td>
</tr>
<tr>
<td>Language Aggregation Strategy</td>
<td>String</td>
<td>Flatten Clusters From Majority Language</td>
<td>Flatten Clusters From Majority Language</td>
<td>Flatten Clusters From Majority Language</td>
</tr>
<tr>
<td>Phrase Document Frequency Threshold</td>
<td>Integer</td>
<td>1-100</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Exact Phrase Assignment</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Merge Lexical Attributes</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Minimum Cluster Size</td>
<td>Integer</td>
<td>1-100</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Reload Lexical Resources</td>
<td>Boolean</td>
<td>True \lor False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>Word Document Frequency Threshold</td>
<td>Integer</td>
<td>1-100</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 9: The parameters for the Lingo algorithm, their type, the range for the values that they can be given, their standard value, and the final value that was set.
D Expert Review Contents

Inleiding
De website die je gaat beoordelen is een zoekmachine gericht op blinde gebruikers. Dit betekent dat de gebruikers van de zoekmachine de webpagina's moeten kunnen bekijken met een screen reader en/of een brailleleesregel. Ook kan er vanuit gegaan worden dat er genavigeerd wordt met een toetsenbord. De website bevat geluid, dus het is handig om tijdens het testen van de website je geluid aan te zetten.
Hieronder volgt eerst een voorbeeldscenario en korte instructies om kennis te maken met de website. Daarna volgen de vragen voor het beoordelen van de website.

Voorbeeldscenario
In de volgende stappen wordt kort omschreven hoe het gebruik maken van de zoekmachine door een gebruiker in zijn werk kan gaan.

Stap 1. De gebruiker gaat naar de beginpagina van de zoekmachine en hoort een ‘plieng’-geluid als de pagina goed geladen is. De focus van de webpagina ligt op het zoekveld.

Stap 2. De gebruiker kan een zoekterm invoeren en met een druk op de ‘Enter’-knop zoeken. Een andere mogelijkheid om te zoeken is door met screen reader commando’s naar de zoekknop (die zich naast het zoekveld bevindt) te gaan en daar op de ‘Enter’-knop te drukken.

Stap 3. De zoekmachine laadt de resultatenpagina waarbij de focus van deze webpagina naar het kopje ‘Eerste resultaten’ springt; nadat de pagina is geladen klinkt er weer een ‘plieng’.


Kennismaking met de website
Kies een screen reader en browser die jouw voorkeur hebben. Zoek voordat je de vragen gaat beantwoorden eerst op de volgende drie zoektermen om zo een idee te krijgen van de website:
Term 1: Appeltaart recept
Term 2: Deep learning
Term 3: Paralympische spelen

Zoek nu op zelfgekozen zoektermen. Doe dit voor minstens 3 verschillende zoektermen en vermeldt hieronder welke dat zijn.
Term 1:
Term 2:
Term 3:

Hierna mag je ook op andere termen zoeken als dit handig is voor het beantwoorden van de vragen of verdere verkenning van de site.

De vragen (ingedeeld in vier groepen):
Hieronder volgen de beoordelingsvragen. Achter de naam van elke groep staat aangegeven hoeveel vragen er zijn. Het totaal aantal vragen is 28. Geef bij het beantwoorden van elke vraag een cijfer voor hoezeer het gevraagde geldt (op een schaal van 1-10) met een korte toelichting.
**Thema 1: Zichtbaarheid van de status van de website (zeven vragen)**

Vraag 1: Is het voor gebruikers duidelijk wanneer de pagina geladen is?
Cijfer en toelichting:

Vraag 2: Is het voor gebruikers duidelijk wat ze op de pagina’s kunnen verwachten?
Cijfer en toelichting:

Vraag 3: Als er geen zoekresultaten gevonden zijn, is dat dan duidelijk voor gebruikers? (Dit is te testen door een heel lang woord van random letters in te voeren als zoekterm.)
Cijfer en toelichting:

Vraag 4: Wordt er op een nuttige manier gebruik gemaakt van korte geluiden?
Cijfer en toelichting:

Vraag 5: Kan het gebeuren dat gebruikers snel fouten maken en zo ja, kunnen ze die fouten goed corrigeren? (Denk hierbij bijvoorbeeld aan het uitvoeren van de zoekopdracht zonder dat ze die hebben afgeschreven.)
Cijfer en toelichting:

Vraag 6: Zijn de foutmeldingen duidelijk geformuleerd?
Cijfer en toelichting:

Vraag 7: Is er hulp-informatie beschikbaar, die makkelijk doorzocht en opgevolgd kan worden?
Cijfer en toelichting:

**Thema 2: De leesbaarheid en bruikbaarheid van de inhoud (acht vragen)**

Vraag 1: Is de inhoud van de pagina’s goed te vinden, te lezen en te begrijpen?
Cijfer en toelichting:

Vraag 2: Is de benodigde informatie op de website aanwezig en is de informatie relevant?
Cijfer en toelichting:

Vraag 3: Staan alle links die nodig zijn op de website en staan er geen overbodige links op de pagina’s?
Cijfer en toelichting:

Vraag 4: Is het duidelijk waar de links naartoe gaan?
Cijfer en toelichting:

Vraag 5: Is de website intuïtief?
Cijfer en toelichting:

Vraag 6: Bevat de website elementen die het mogelijk maken voor meer ervaren gebruikers om taken sneller uit te voeren?
Cijfer en toelichting:

Vraag 7: Is het duidelijk voor de gebruiker welke input ze moeten geven als er input van ze verwacht wordt?
Cijfer en toelichting:

Vraag 8: Is het zoekveld makkelijk te vinden op de pagina en goed te gebruiken?
Cijfer en toelichting:

Thema 3: De structuur van de website (vijf vragen)
Vraag 1: Zit er een duidelijke structuur in de website? (Denk hierbij aan het gebruik van duidelijke kopjes, het scheiden van verschillende elementen en een logische volgorde.)
Cijfer en toelichting:

Vraag 2: Is het duidelijk wat je kan verwachten bij elk onderdeel van de website? (Denk bij onderdelen bijvoorbeeld aan het zoekgedeelte en de resultatengedeeltes.)
Cijfer en toelichting:

Vraag 3: Zijn de gebruikte anchor links nuttig en duidelijk? (Een voorbeeld van zo’n link is het verspringen van de focus op de website naar de resultaten als je gezocht hebt.)
Cijfer en toelichting:

Vraag 4: Staan er niet teveel of te weinig titels/kopjes op de webpagina’s?
Cijfer en toelichting:

Vraag 5: Is de functie van de onderdelen van de website op de verschillende pagina’s altijd hetzelfde?
Cijfer en toelichting:

Thema 4: Gebruik met een screenreader (acht vragen)
Vraag 1: Is de naamgeving van verschillende onderdelen en elementen duidelijk? (Denk hierbij aan titels, de naam van het zoekveld, et cetera.)
Cijfer en toelichting:

Vraag 2: Zijn er alternatieve teksten beschikbaar waar deze nodig zijn?
Cijfer en toelichting:

Vraag 3: Zijn er storende veranderingen in de website waardoor de gebruiker onderbroken of gehinderd wordt tijdens het gebruik? (Denk hierbij bijvoorbeeld aan het herladen van de pagina op ongewenste momenten of het ongewenst verspringen van de focus.)
Cijfer en toelichting:

Vraag 4: Is de website te gebruiken met verschillende soorten webbrowsers en screen readers?
Cijfer en toelichting:

Vraag 5: Is het makkelijk en overzichtelijk om de website te navigeren?
Cijfer en toelichting:

Vraag 6: Is de lay-out van de pagina niet storend voor de functionaliteit?
Cijfer en toelichting:

Vraag 7: Zijn de lay-out en de gebruikte termen hetzelfde op de verschillende pagina’s?
Cijfer en toelichting:

Vraag 8: Is er op een goede manier gebruik gemaakt van HTML en CSS?
Cijfer en toelichting: