TU WIEN Informatics

Merging the Gap Between Automated and Human Centered Usability Testing

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Can Özgür Yilmaz, BSc
Matrikelnummer 01029434

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dr. Allan Hanbury
Mitwirkung: Univ.Ass. Mag.rer.nat. Dr.techn. Julia Neidhardt

Wien, 30. August 2022

Can Özgür Yilmaz	Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Merging the Gap Between Automated and Human Centered Usability Testing

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Can Özgür Yilmaz, BSc
Registration Number 01029434

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dr. Allan Hanbury
Assistance: Univ.Ass. Mag.rer.nat. Dr.techn. Julia Neidhardt

Vienna, 30th August, 2022

Can Özgür Yilmaz	Allan Hanbury

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Can Özgür Yilmaz, BSc

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

Wien, 30. August 2022

Can Özgür Yilmaz

Danksagung

Ich möchte mich bei dem Professor dieser Arbeit Univ.Prof. Dr. Allan Hanbury, meinem Betreuer Univ.Ass. Mag.rer.nat. Dr.techn. Julia Neidhardt für ihre Arbeit und ihr wertvolles Feedback zu dieser Arbeit, Katharina Wünsche BSc. und Seungbin Yim MSc. für ihre hervorragende Arbeit am DYLEN Tool und ihr wertvolles Feedback zu SAUTO, Asil Cetin MSc. dafür, dass er mich auf die Idee für diese Arbeit gebracht und mich in das DYLEN Projekt eingeführt hat, Dalibor Pančić BSc. für seine außergewöhnliche Arbeit bei der Einrichtung der Infrastruktur für das DYLEN Tool und SAUTO, allen Teilnehmern der manuellen Usability-Tests, allen Nutzern, die das DYLEN Tool nach dem Launch ausprobiert haben, Anıl Aktan und Lale Tüver B.A. für ihre Hilfe bei der Suche nach Nutzern für die summative Phase und schließlich Johannes Faas MSc. für das Korrekturlesen der deutschen Übersetzungen bedanken.

Acknowledgements

I would like to thank the Professor of this thesis Univ.Prof. Dr. Allan Hanbury, my supervisor Univ.Ass. Mag.rer.nat. Dr.techn. Julia Neidhardt for all her work and valuable feedback on this thesis, Katharina Wünsche BSc. and Seungbin Yim MSc. for their exceptional work on the DYLEN tool and their valuable feedback on SAUTO, Asil Cetin MSc. for providing me with the idea for this thesis and introducing me to the DYLEN project, Dalibor Pančić BSc. for his exceptional work on setting up the infrastructure for the DYLEN tool and SAUTO, all the participants of the manual usability tests, all the users who tried out the DYLEN tool after launch, Anıl Aktan and Lale Tüver B.A. for helping me find users for the summative phase and finally Johannes Faas MSc. for proof reading German translations.

Kurzfassung

Heutzutage ist Usability Testing ein wesentlicher Aspekt des Softwareentwicklungsprozesses. Konventionelle Usability Tests sind jedoch sehr zeit- und ressourcenintensiv, da eine reale Person den Benutzer beobachten und versuchen muss, anhand ihrer Beobachtungen Usability Probleme zu entdecken. Als Lösung für dieses Problem stellen wir ein Framework für semi-automatisierte Usability Tests vor, bei dem der Benutzer die Anwendung ohne Beobachter testet und anschließend seine Interaktionen erfasst, gespeichert und dann analysiert werden, um potenzielle Usability Probleme aufzudecken. Unser vorgeschlagenes Framework kombiniert vier allgemeine semi-automatisierte Usability Testansätze zu einem: Metrik-basiert, Mustervergleich, Aufgaben-basiert und Inferential. Um unser vorgeschlagenes Framework zu evaluieren, haben wir ein semi-automatisierte Usability Tool (SAUTO) entwickelt und während der Entwicklung einer Webanwendung, dem DYLEN-Tool, manuelle und semi-automate Usability Tests durchgeführt. Wir haben das Framework evaluiert, indem wir die Ergebnisse manueller Usability Tests mit semi-automatisierte Usability-Tests verglichen und auch qualitative Interviews mit den Entwicklern von DYLEN und einem externen Usability Experten geführt haben. Die Ergebnisse zeigen, dass semi-automatisierte Usability Tests manuelle Usability Tests zwar nicht ersetzen können, es aber sehr sinnvoll sein kann, beides zusammen durchzuführen. Darüber hinaus ermöglicht semi-automatisierte Usability Testen das Testen einer Software mit einer großen Anzahl von Benutzern, was mit manuellen Usability Tests nicht möglich ist. Schließlich ermöglicht es die Durchführung von Usability-Tests nach dem Launch einer Software.

Abstract

In today's world, usability testing is an essential aspect of the software development process. Conventional usability testing however is very time and resource consuming, because there needs to be a real person observing the user and trying to discover usability problems from their observations. As a solution to this problem, we present a framework for semi-automated usability testing, in which the user tests the application without an observer, and then their interactions are captured, saved and then analysed to discover potential usability problems. Our proposed framework combines four general semi-automated usability test approaches into one, these are: metric based, pattern matching, task based and inferential. To evaluate our proposed framework we developed a semi-automated usability tool (SAUTO) and conducted manual and semi-automated usability tests during the development of a web application, the DYLEN tool. We evaluated the framework by comparing the results of manual usability tests to semi-automated usability tests and also by conducting qualitative interviews with the developers of DYLEN and an external usability expert. The results indicate that while semi-automated usability testing cannot replace manual usability testing, it can be very useful to carry out both together. Additionally, semi-automated usability testing allows to test a software with a high amount of users which is not possible with manual usability testing. Finally it allows carrying out usability tests after the launch of a software.

Contents

CHAPTER 1

Introduction

Usability of a software application is one of its most important aspects. It is often usability that makes or breaks an application, such that even if it is superior in other aspects like underlying technology, without efficient usability, the application will most probably fail. In other cases usability problems can produce disastrous results. For instance, a newly implemented feature in Boeing 737 MAX was the root cause of two fatal crashes [1]. This feature took control away from the pilots in certain cases and did not provide any feedback. In one crash, the pilots were unaware of the feature and in the other, they did not know how to turn it off. If this feature provided feedback to the pilots, the crashes could have been avoided. Another example for faulty usability causing fatal disasters was the case of Therac-25, a computer-controlled radiation therapy machine. Among other problems, the machine displayed only error codes without any explanations and did not allow users to remedy the errors. This lead to the system being in a state, of which the user (machine operator or health personnel) was not notified [2]. In the end, the machine was involved in at least six accidents between 1985 and 1987. Such disasters prove that usability testing is an essential part of the software (or any product) development process in order to find out possible usability problems and improve the user experience.

Usability testing in software is conventionally carried out with at least one user who doesn't have previous experience with the tested software, an observer (moderator) and a device to test on [3]. The user is expected to complete tasks using the software, while the observer watches to discover the problems that the user encounters. Tests can be carried out remotely via video call software or telephone. The user's interactions can be recorded and observed at a later date. We call these kinds of usability testing "manual", as in an observer observes and manually finds usability problems or possible improvements.

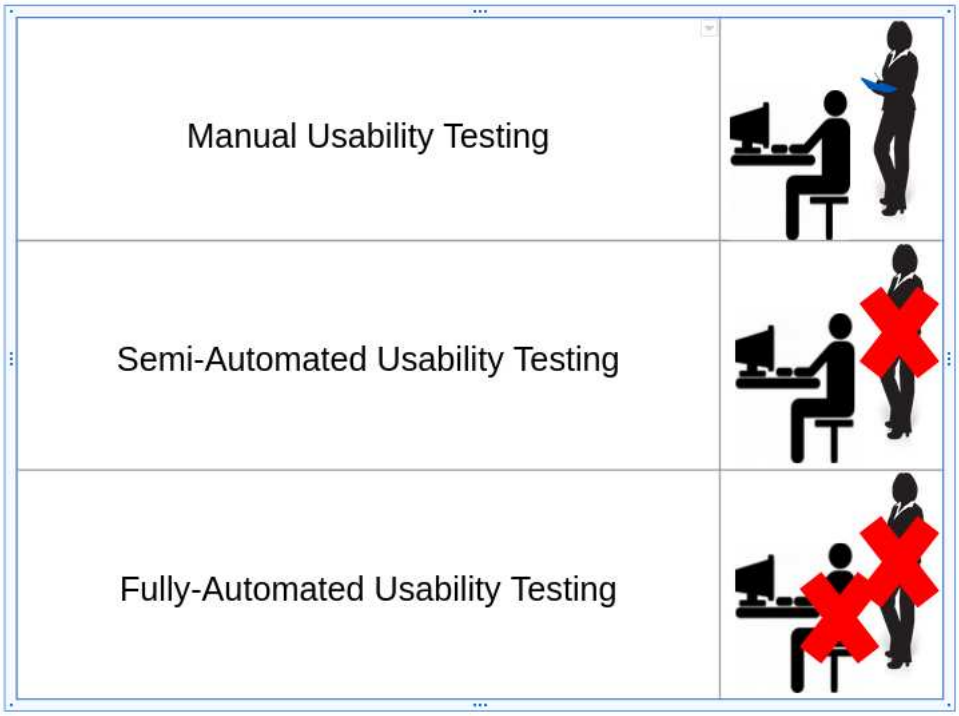

Figure 1.1: Usability Testing Automation Types

1.1 Motivation and Problem Definition

Conventional usability testing is a an expensive and lengthy process. The users need to be gathered (sometimes physically), the developers need to take their time to carry out the usability tests. The results need to be evaluated. And most importantly, these tests are almost always carried out in a small scope i.e. with only a small number of users, because of time and resource constraints. Therefore, there has always been an opportunity to improve and optimize this process via automation to make it more efficient.

We define three usability testing automation types: manual usability testing, semi-automated usability testing and fully-automated usability testing. They are described in the following and can be viewed in Figure 1.1.

Ivory et al. [4] define three types of automation on usability testing: *automatic capture, automatic analysis and automatic critic*. Conventional usability testing (manual usability testing) requires none of these types, as everything is done manually by the observer. However, since manual usability testing requires a lot of time and resources, there has been much research done in way of automating this process. Firstly, tools have been developed to record usability testing sessions, visualize results and overall enhance the usability testing process. These tools can be considered to provide *automatic capture*. Some examples of these tools are discussed in the chapter State of the Art. They range from analyzing eye movements to analyzing feedback questionnaires, from tracking mouse

movements to recording the test to be able to replay it again in order to analyze it. In short, they automate some aspect of usability testing and therefore enhance the usability testing process, however they can't be called automated usability testing, as there is still a need for a human observer to manually analyze the tests.

Secondly, there has been research into *automatic analysis*, which analyzes testing session logs after the test has been carried out and automatically detects usability problems. If *automatic analysis* tools can suggest improvements, they become *automatic critic* tools. We define this automatic log analysis (and critic) as "semi-automated" usability testing. There is still a need for a user to complete tasks but no observer is necessary. It is worth noting that these session logs are separate from regular logs that an application might produce. These are independently gathered user interaction data during the session specifically for usability testing purposes. The great advantage of semi-automated usability testing is that there is no need for an observer. This saves precious time for developers and allows usability testing on a larger scale.

Finally, there has been some research into tools that detect usability problems automatically without a user. These tools apply pre-defined usability guideline models on the tested software to generate usability predictions, which Ivory et al. [4] define as *analytical modeling*. They also use models to mimic a user interacting with an interface, which Ivory et al. define as *simulation*. We define this as "fully-automated" usability testing, because there is no real user involved. Tools that can automatically crawl a software application and produce meaningful usability related analysis is a strong idea but Ivory et al. claim that subjective measures such as user satisfaction are unlikely to be predictable by automated methods. Usability testing without human users cannot be human centered, so it defeats the purpose of usability testing. There are also predefined UI design rules, that developers can check against, but these cannot replace tests with real users.

This thesis focuses on semi-automated usability testing. As will be described in the chapter State of the Art, there have been some studies done into semi-automated usability testing with promising results. Tools developed during these studies have shown to produce comparable results to manual testing [5, 6, 7, 8, 9]. There are four kind of approaches to semi-automated usability testing as defined by Ivory et al. [4]. The tools developed in the scope of the previously mentioned research, which will be presented in the chapter State of the Art, only make use of one or two of these approaches. This work proposes a framework that combines all four of the defined approaches in an effort to improve on the existing semi-automated usability testing research.

In the scope of this master's thesis, we developed a novel semi-automated usability testing tool (SAUTO) for the DYLEN web application (see next section), which combined the four previously mentioned semi-automated usability testing approaches. The underlying principles of the tool can be regarded as a framework for semi-automated usability testing of software applications. This framework can be considered our contribution to the field, because combining the four approaches hasn't been done before to our best knowledge. With the developed tool, we show that, while manual usability testing during development

can produce more detailed results than semi-automated usability testing, the time and resource advantages semi-automated usability testing provides during development and the fact that it can be conducted after product launch and with many more users, makes it a valid option for as complementary to manual usability testing. While developing the aforementioned tool, we not only gained insights into advantages of semi-automated usability testing but also experienced the obstacles and pitfalls for it. This work discusses this experience and the pros and cons of semi-automated usability testing in the context of web applications as compared to manual usability testing.

1.1.1 Research Questions

In this work we tried to answer the following research questions:

a. Which resource and time consuming parts of manual usability testing can be eliminated via automation?

b. How effective is semi-automated usability testing compared to manual usability testing?

1.2 Case Study: DYLEN

The DYLEN project[1] bridges the fields of linguistics, digital humanities and computer science in order to explore large-scale authentic language data of the Austrian German Language. Within the DYLEN project, a web application called the DYLEN tool[2] was developed to visualize the semantic change of the Austrian German lexicon (vocabulary) over 20 years. Two sources of data were used to generate linguistic data: the Austrian Media Corpus (AMC) and the Corpus of Austrian Parliamentary Records (ParlAt). AMC covers the entire Austrian media landscape of the past 20 years and contains 40 million texts (more than 10 billion tokens). ParlAT contains the parliamentary records of the National Chamber (Nationalrat), which is the national level chamber of the two chambers in the Austrian parliament, the other one being the Federal Chamber (Bundesrat).

The DYLEN tool was chosen for the case study because of a variety of reasons. First and foremost, the project was already planned and the plan included manual usability testing and the resources for it were already decided and available. Secondly, the project was developed at the Austrian Centre for Digital Humanities, where we already had a work relationship. Finally, the tool itself was planned to be a web application and we had experience with the underlying technologies.

[1] https://dylen.acdh.oeaw.ac.at
[2] https://dylen-tool.acdh.oeaw.ac.at/

1.3 Methodology

The first step in any academic work is the research to the state of the art. We started by looking into the academic literature about the current state of automation of usability testing. Our findings can be found in the chapter State of the Art. We found that there has been research into automating the usability testing process. Ivory et al. has gathered a list of four approaches used in this research [4]. We were able to find published academic papers that automated usability testing using only one or two of these approaches.

After researching into the state of the art, we defined the contribution goal of this master's thesis work to the field: Using all four approaches defined by Ivory et al. in developing a novel semi-automated usability testing framework and showing that it is able to be used in usability testing and can produce meaningful usability testing results.

We chose the DYLEN project as a case study to test the proposed novel semi-automated usability testing framework. In the scope of the DYLEN project, a web application was developed: the DYLEN tool. The project and the development involved user-centered design principles and iterative usability testing as defined (and highly encouraged) by Barnum in their book [3]. Since usability testing was such an integral part of the project, it was a great fit for this work. We took part in the planning and development of the DYLEN tool to see what kind of functionalities the tool is supposed to provide and what kind of user interactions it requires.

During the development of the DYLEN tool, we developed our framework in the form of SAUTO, which was developed to implement the four approaches on automating usability testing defined by Ivory et al. It was developed specifically geared towards the DYLEN tool. Even though the four approaches were implemented generally, SAUTO can not directly be used with another web application, which could be achieved with more work but it was out of the scope of this thesis.

In order to test and evaluate the proposed framework, we conducted the already planned manual usability tests together with semi-automated usability tests. These tests were done in three phases as a means of iterative testing. After about four months of development, we started with the first phase of tests which is the formative phase. We conducted manual usability tests with five testers. At the same time SAUTO captured these sessions and analyzed the user interactions. After the tests we collected our observations from the manual usability tests and the results of the semi-automated usability tests and decided on what to fix/implement next. Next, we repeated this whole process in the next month as the iterative phase, in which we conducted manual usability tests with six testers. Finally, with the feedback from the iterative phase, we finished development of the tool and launched it. This is where the last phase which is the summative phase started, where SAUTO captured 112 sessions in the next two months from usages of the tool by real users. The following is a list that summarizes the the phases:

- **Formative phase:** 5 users - manual and semi-automated usability tests

- **Iterative phase:** 6 users - manual and semi-automated usability tests
- **Summative phase:** 112 users - only semi-automated usability tests

1.3.1 Evaluation

After the tests were finished, we evaluated the results and our research questions in two ways: quantitatively and qualitatively. All three research questions are answered in the chapter Conclusion. To quantitatively evaluate our proposed tool and to answer, if semi-automated usability testing can be as effective as manual usability testing, we first compared the number of usability problems found during manual usability tests and semi-automated usability tests with each other in the formative and iterative phases. We additionally compared the number of usability problems found in the summative phase (only semi-automated) to the first two phases. Since we could not compare the number of usability problems found in the summative phase with a manual test like in the first two phases, we used another approach for evaluation: qualitative interviews.We interviewed the two developers of the DYLEN tool and an external usability expert. The interviews were standardized open-ended, which means that all interview partners received the same open ended questions and were encouraged to elaborate their answers. The answers were compiled into themes and are presented in the chapter Conclusion.

1.4 Overview

This thesis is organised as follows. In the chapter Introduction, we discussed the problem we tackled and our motivation along with our methodology in this thesis. In the chapter State of the Art, we present research on conventional usability testing, research on enhancing conventional usability testing via automation and finally research on semi-automated usability testing. In the chapter DYLEN, we present the DYLEN project and the DYLEN tool. In the chapter Approach, we describe the case study we used to test our proposed framework, the implementation of it and our evaluation approach. In the chapter Phases, we present the three phases of the usability testing case study, the results of the usability tests and the interviews we conducted with experts about our proposed framework. Finally, in the chapter Conclusion, we conclude this thesis by discussing the results and evaluation.

CHAPTER 2

State of the Art

In the previous chapter, we described usability testing, and the automation of it. In the following, we look into the academic background of it by discussing research into conventional (manual) usability testing, enhancing the usability testing process and finally semi-automated usability testing. When selecting the works to present in this chapter, we applied different criterion for our research. Works about usability testing in general were chosen because they aimed to describe all of the concepts from the field of usability and usability testing. For the works about enhancing conventional usability testing, we tried to filter out works older than 20 years and finally for the works about semi-automated usability testing, we included an exhaustive list of works to the best of our knowledge.

2.1 Conventional Usability Testing

Usability Testing has been researched extensively, especially for software in the last three decades. Its use in the industry has led to a great amount of experience being gathered. All the research and practical experience points to usability testing being essential in the software development process.

Barnum discusses every aspect of conventional usability testing in the book *Usability Testing Essentials Ready, Set...Test* [3]. In the book, they point out that usability is invisible, meaning that users don't think about it when it is done well. They pay attention to it only when it is bad and they don't tolerate it. That's why they argue that it is essential in every software and stakeholders must be convinced of its uses as early in the project as possible. Furthermore, they define usability, usability testing and all its related terminology, they explore available tools when working with usability and user experience, they talk about understanding the users and their goals, they go over how to plan, prepare and conduct usability testing with best practices in mind, and finally they outline how to analyze the findings of usability testing, how to make sense of them and

how to report them to stakeholders and developers. We planned, prepared, conducted and analyzed the manual usability tests following the best practices described by Barnum. We also chose the usability test parameters like remote vs live based on the project resources and the recommendations given by Barnum in their book.

Barnum defines user centered design as the process of building insights about users' experience through usability testing and other forms user research into product development through an iterative design process [3]. Furthermore, Barnum advocates for follow-up studies, because small one time user studies typically show where the problems are, but not what the solutions are. Having follow-up user studies is called iterative testing [3]. For both manual and semi-automated usability testing of the DYLEN tool, we adopted user centered design and iterative testing.

Aamir et al. have conducted a research survey over usability testing of web applications [10]. In their paper, they summarize literature about the benefits of good usability. They categorize web sites on the basis of their functionality into three: information oriented(the DYLEN tool), service oriented, business oriented. They claim that these categories might affect the design rules and design specifications. For example a a business oriented website might focus on ordering a product, while an information oriented website might focus on text being easily readable. We focused on the design specifications of the DYLEN tool in this information oriented context. Aamir et al. also go over usability evaluation methods in research. From the evaluation methods they list, we used two evaluation methods defined by Adelman et al. [11] when conducting manual usabilty tests: the empirical method, which is based on user actions and the subjective method, which is based on user opinions. We didn't employ the heuristic method, which is based on expert opinion. Furthermore, they detail usability testing methods that are used in conventional usability testing. Finally, they propose a usability testing method called CARE, which tries to eliminate time taking factors and expensiveness of other techniques.

Usability testing can be done once or in multiple phases during software development. If it is planned to carry out usability tests more than once (relevant for user centered design), the testing can be divided into separate phases: formative, iterative and summative. Formative is done at first to form initial ideas about user interactions with the software. Then, after fixing the discovered problems in the formative phase, developers can conduct usability testing in the iterative phase and repeating the fix and test process as much as they want/can. West et al. explain that formative and iterative usability studies focus on identifying usability problems during the initial part of development before the product is fully shaped [12]. They focus less on metrics and focus more on qualitative feedback and participant/observer interaction, which is what manual usability testing provides. West et al. define a third usability study as summative (also called metric or benchmark study), which is done at the later stages of the development towards the end or even after product launch. In it, more value is put on quantifying performance data of the product.

2.2 Enhancing Usability Testing

Before we discuss semi-automated usability testing, we would like to take a look at research into automating and enhancing the usability testing process. While semi-automated testing aims to (at least partially) get rid of the observers, enhancing and automating the usability testing process aims to help moderators during conventional usability testing. This research can help get a better understanding of and provide insights into usability testing and it can also be regarded as a stepping stone to semi-automated usability testing. Starting point of all the following research is that conventional usability testing takes too much time and resources. Therefore, they propose various solutions to improve the efficiency of it.

Filho et al. bring a unique aspect to usability testing by analyzing live emotions through facial expressions during user interaction in mobile applications [13]. They capture the facial expressions of users through the mobile phone's camera during usability testing and they successfully connect users' emotions with their interactions with the mobile applications. However, they do not combine this valuable data with usability issues discovered via automated log analysis (i.e. semi-automated usability testing), which is a promising aspect for future work.

West et al. focus on conducting questionnaires after usability testing [12]. They captured success rate, duration and user given grade per task. They demonstrate that the results of the tests and questionnaires were comparable to the results gathered by a usability expert, who observed users during usability testing (i.e. conducted conventional usability testing).

Gonzalez et al also focus on questionnaires after usability testing [14]. They use data mining on the answers to questionnaires to get a quantitative usability evaluation from qualitative evaluations. They tested 69 websites and collected 3450 answers to analyze. They employ association rules and decision trees on the data set of answers. The questionnaires were filled after users tested selected websites. The results give insights into what aspects and properties of websites affect usability the most.

El-Halees goes in a similar direction and presents a solution based on opinion mining of test subjects' statements after usability testing is completed [15]. They propose a model that evaluates the statements on three aspects: effectiveness (i.e. can users do what they want to do) , efficiency (i.e. how much effort do users require to do their task) and satisfaction (i.e. what do users think about the software's ease of use). They conclude that the proposed model can successfully evaluate software's subjective usability.

Jeong et al. propose a framework and a tool which visualizes captured user actions in an easy way to the conductors of the usability tests [16]. Their tool is especially helpful when testing a large number of interaction data. Thus developers can spend much less time to conduct usability tests and can focus more on captured data from the tests.

Fabo et al. argue that semi-automated usability testing only focuses on the events generated by the user but not the user himself/herself [17]. To mitigate this, they

introduce a tool which captures and analyzes user interactions and also tracks users' eye movements real time. Additionally, the tool takes a screenshot every time a user interaction event occurs. After the tests, the tool presents statistics and a live replay functionality to help visualize results.

Ferre et al. propose an extension on Google Analytics for Mobile to capture and log low level user events [18]. With their proposed extension tool, there is no need to have a real observer keeping time of user actions. It is important to note that the observer they eliminate with their extension is only responsible for keeping track of the time and not analyzing usability problems. Therefore this paper helps with enhancing manual usability testing and can not be considered semi-automated usability testing. Ferre et al. state that the tool can be used in controlled environments like during lab usability testing and control free environments like after launch with real users. They analyze the gathered user data by comparing user action sequences to the optimal path previously defined by developers.

As it is clear from the research described above, manual usability testing is a time and resource consuming process and automating parts of it has been a goal of researchers. However, the above described papers help improve manual usability testing and do not automate the process itself. The tests are still defined as manual because there is still a need for an observer during (or after via video recordings) tests.

2.3 Semi-Automated Usability Testing

In this section we discuss the literature on semi-automated usability testing. Firstly, we go over a survey that identifies four different approaches to semi-automated usability testing, then we discuss papers that employ one or more of these approaches.

Ivory et al. surveys the state of the art on automating usability evaluation [4]. They cite the survey done by Balbo et al., who identify four approaches to automation [19]:

1. Non-automatic: methods performed by humans
2. Automatic Capture: methods that rely on software facilities to record relevant information about the user and the system
3. Automatic Analysis: methods that are able to identify usability problems automatically
4. Automatic Critic: methods that not only point out difficulties but propose improvements

Semi-automated usability testing requires on automatic capture and automatic analysis with little to no input from developers and testers. Automatic capture is relatively simple compared to automatic analysis. Automatic critic can also be included in semi-automated

usability testing. Ivory et al. identify four general approaches for automatic analysis. They call it: analyzing log files generated from user interactions, which we define as semi-automated usability testing. These four approaches are as follows:

1. **Metric-based** approaches generate quantitative performance measurements like task completion time, and productive period (i.e., portion of time the user did not have problems), command frequency, the number of physical operations required to complete a task, and required changes in the user's focus of attention on the screen
2. **Pattern-matching** approaches which analyze user behavior captured in logs, detects and reports repeated user actions that may indicate usability problems.
3. **Task-based** approaches analyze discrepancies between the designer's anticipation of the user's task model and what a user actually does while using the system.
4. **Inferential** analysis of Web log files includes both statistical and visualization techniques. Statistical approaches include traffic-based analysis (e.g., pages per visitor or visitors per page) and time based analysis (e.g., click streams and page-view durations) and visualization includes mouse click or mouse movement hotspot heat maps

To the best of our knowledge, there has been no studies, where these four approaches were all used together in one. In this thesis, we combine these four approaches in one, in our framework. The following papers are about tools that make use of only one or two of these approaches.

Baker et al. propose an analyzer software which compares the expected and the actual use of a user interface during usability testing [5, 6]. This tool provides comparison checking, assertion checking, and hotspot analysis:

1. Comparison analysis indicates the extent of similarity between how a developer thinks an application's functionality should be accessed, as expressed in an expected action sequence description, and how a user actually accesses it, described by an actual action sequence.
2. Assertions are checks that evaluate to either true or false in a similar fashion to that of unit tests. Assertions use a predefined set of usability metrics as thresholds for certain aspects of usability. These thresholds are set by developers according to the usability guidelines they aim to follow.
3. Hotspot analysis presents a color map portraying the relative use of each component for a given type of interaction, such as a mouse click or text input.

Comparison analysis and assertion analysis correspond to the task-based approach. Hotspot analysis corresponds to inferential analysis, because of its visualization nature.

Kluth et al. discuss a semi-automated usability testing tool for iOS applications [7]. Their tool analyzes user interaction sequences and detects usability problems based on design pattern recognition. This corresponds to the pattern-matching approach defined above. They propose to use this tool with real users after the app is launched to get real data, instead of data from a controlled environment.

Lettner et al. propose a framework to log android application interactions [8]. They propose static and dynamic metrics to evaluate the usability of an application. Static metrics compare the structure and design of the application to existing style guides and therefore don't involve the user. Dynamic metrics are extracted from user interaction logs. The data may be used to estimate and determine the application's bounce rate (i.e. a user visits and immediately leaves a screen due to a navigation error), the number of screen calls per visit, the dwell time per screen and the application's conversion rate (e.g. how many visitors convert into regular users). These dynamic metrics correspond to inferential based approach.

Bures et al. propose a tool which scans smart TV UI, and creates a model, with which the developers can define test scenarios for their applications. The users then test the application, during which user interactions are captured and compared to the defined test scenarios [9]. This kind of testing matches the definition of task-based approach exactly.

Grigera et al. define certain usability smells and propose a tool, with which these usability smells can be detected by analyzing user interactions from session logs [20]. Usability smells are defined by Grigera et al. as hints of usability problems. This approach corresponds to the task-based approach, because they define negative expectations and see if user interactions match them.

Semi-automated usability testing is a usability testing method, where an automated tool captures and analyzes user interaction data. It contains four different approaches: metric-based, pattern matching, task-based and inferential. We described above the research that applied one or two of these approaches. We combine these four approaches in our framework to get the most extensive results in terms of semi-automated usability testing.

CHAPTER 3

DYLEN

The DYLEN project[1] bridges the fields of linguistics, digital humanities and computer science in order to explore the diachronic dynamics of lexical networks on the basis of large-scale authentic language data. What makes the language data authentic is that it is taken from real life usage of the language. The project exploits language data that is already available at the Austrian Centre for Digital Humanities and Cultural Heritage (ACDH-CH), namely the Austrian Media Corpus (AMC) and the Corpus of Austrian Parliamentary Records (ParlAT). The corpora represent a part of the Austrian German language, because they contain data from news publications and parliamentary discussions. In the following section, the corpora are described in detail.

From linguistic perspective, the project explores the diachronic dynamics of lexical networks and discuss networks-based methods for diachronic linguistics. From the point of view of computer science, the project applies network theory to a large amount of diachronic linguistic data and discuss new methods for automatic analysis and comparison of these networks. Furthermore, the project enriches the already available digital toolbox with a freely available tool for network analysis and visualisation and will enhance already existing data with additional annotations.

The project was funded by the ÖAW go!digital Next Generation grant (GDNG 2018-020) and was conducted by an interdisciplinary team from the ACDH-CH, the University of Vienna and TU Wien.

3.1 Sources of Data

As previously mentioned, two corpora were used to generate linguistic network data for the DYLEN tool.

[1] https://dylen.acdh.oeaw.ac.at

The AMC covers the entire Austrian media landscape of the past 20 years and contains 40 million texts (more than 10 billion tokens). In the tool AMC is divided into sub-corpora to simplify querying. The available sub-corpora are of singular newspapers and magazines in the period from 1996 to 2017.

- Falter
- Kleine Zeitung
- Kurier
- Die Presse
- Profil
- Der Standard

There are also three sub-corpora of all of them together.

- All magazines
- All newspapers
- All magazines and newspapers

The ParlAT corpus contains the parliamentary records of the National Chamber (Nationalrat), which is the national level chamber of the two chambers in the Austrian parliament, the other one being the Federal Chamber (Bundesrat). ParlAT consists of the official shorthand transcripts from 1996 to 2017 and contains more than 75 million tokens. Besides being linguistically annotated (part-of-speech tagged and lemmatized), the corpus also contains semantic annotations for example all speakers are identified and marked up.

3.2 DYLEN Tool

The DYLEN tool[2] was chosen as the subject web application for this master's thesis, i.e. manual and semi-automated usability testing was conducted on the DYLEN tool. Therefore it is necessary to describe the tool, how it works, what features it has, what it provides and its purpose. The following descriptions of the tool, its components and specifically descriptions of linguistic relevant terms were originally written by the DYLEN team[3] and adapted to this work. They can also be found on the info page (starting page) of the DYLEN tool[4].

[2] https://dylen-tool.acdh.oeaw.ac.at/
[3] primarily by Mag. Klaus Hofmann BA and Anna Marakasova MSc
[4] 2

The DYLEN tool describes itself as an interactive visualization tool, which helps users to gain insight into the dynamics of the Austrian German lexicon (vocabulary) over twenty years in the recent past and to measure semantic change, based on two different text corpora, AMC and ParlAT. It offers visualizations of three different network types:

- **Ego Network:** This network type allows the user to explore the semantic neighborhoods of target words in different years
- **General Network (Party):** This network type allows the user to explore lexemes[5] used by political parties in the Austrian Parliament
- **General Network (Politicians):** This network type allows the user to explore lexemes used by individual speakers in the Austrian Parliament

3.2.1 Ego Network

Ego network is a graph that visualizes a selected target word to provide information about its semantic neighborhood in a given year. The nodes in the network represent the top 50 lexemes (or words or word goups) that are most closely related to the target word semantically. An example is shown in Figure 3.1.

Users can select Ego Network from the top navigation. Then they can select their preferred corpus and sub-corpus from the drop-down menus. They can then choose a target word by typing into the text input and using the auto-complete function. Finally, when they click on visualize, they will be able to view the ego network visualization.

The target word itself is not displayed in the visualized ego network, since by definition, every node in the network is connected to the target word by an edge. The size of a node represents its word frequency. The thickness of the edges represent the semantic similarity between nodes. The colors of the labels represent different part-of-speech tags (noun, verb, adjective, proper noun). The colors of the nodes represent semantic clusters, which may be interpreted as polysemic meanings (one word can have more than one meaning) or distinct usage contexts, it can be enabled by clicking on the “show clusters” checkbox.

Users can use the the ego network visualization to:

- View the change of usage of a target word over time by using the year slider
- Visualize possible distinct meanings or usages by clicking on the “show clusters” checkbox
- Identify which meaning or usage of the word was dominant by comparing cluster sizes

[5]Definition of lexeme according to Oxford dictionary: a word or several words that have a meaning that is not expressed by any of its separate parts

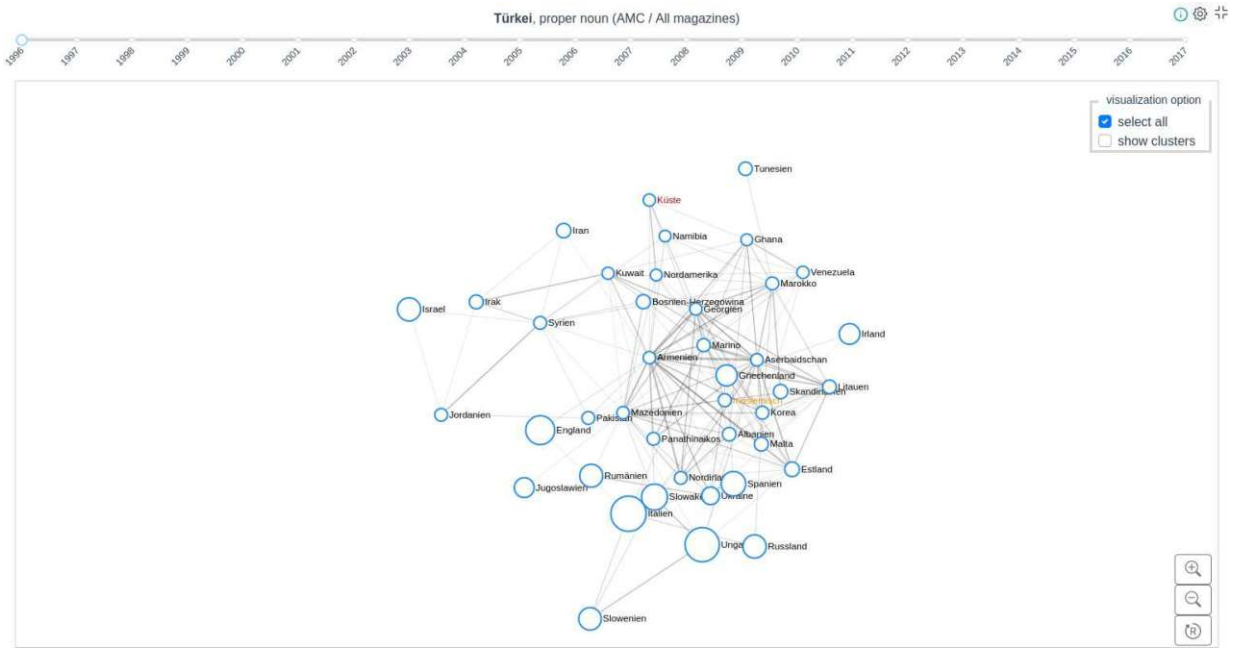

Figure 3.1: Ego Network Visualization of the word **Türkei** in the year 1996 in all magazines

- Compare two target words by visualizing two networks in parallel
- Observe the similarity between words by comparing the thickness of the edges
- Set a node as a target word and search for it by right clicking on it
- See exact values of a node by right clicking on it

3.2.2 General Network

General Network is a graph that visualizes a selected party or speaker in parliament to provide information about their discourse topics in a given year. The nodes in the network represent the most frequent lexemes used by the selected party or the selected speaker. An example is shown in Figure 3.2.

Users can select either General Network (Party) or General Network (Speaker) from the top navigation. They can then select a party and a speaker (only if they chose speaker from the top navigation) from the drop-down menus. Before they visualize the network, they need to filter it down because general networks are very large in size, and visualizing the network is computationally expensive. The filtering is done by choosing the node metric that users want to filter with and choosing a percentage range. Finally, they can click on visualize to view the general network. If the filter range is too big, the DYLEN

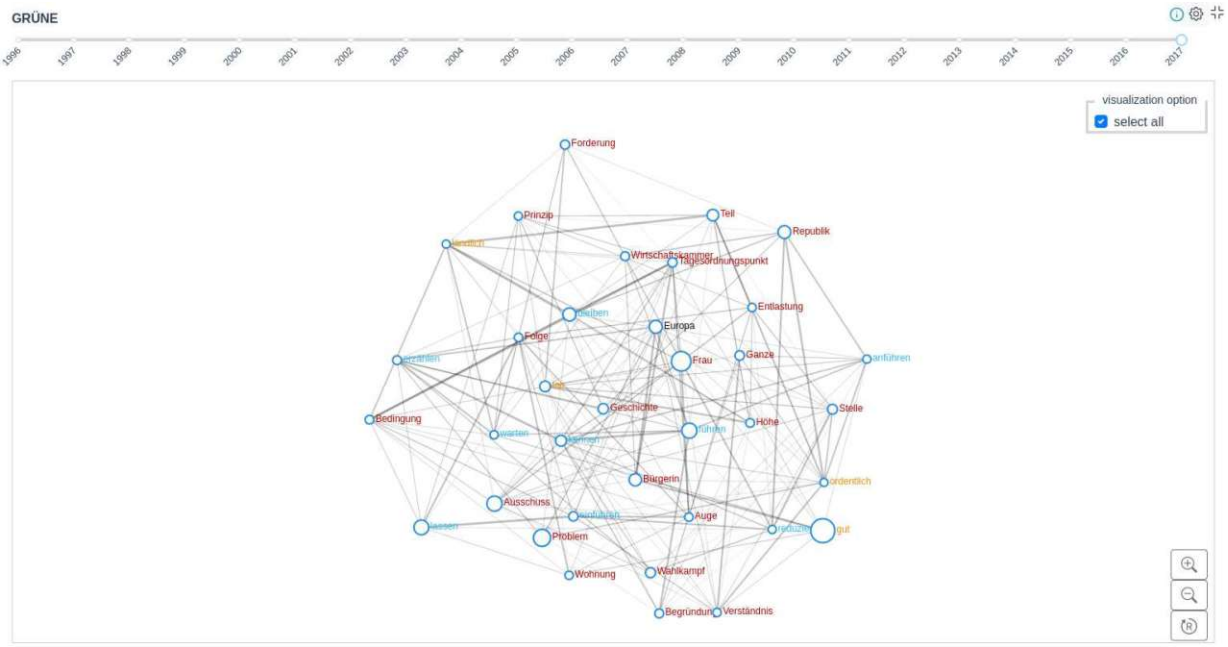

Figure 3.2: General Network Visualization of the party **Grüne** in the year 2017 from ParlAT with Degree Centrality between 0% and 5%

Tool might cancel the query and show an error asking the user to select a smaller range. This is done to prevent freezing the user's browser.

Similar to Ego Network, the size of a node represents its word frequency. The thickness of the edges represent the semantic similarity between nodes. The colors of the labels represent different part-of-speech tags.

Users can use the the general network visualization to:

- view what parties focus on in their speeches by using the General Network (Party) visualization.
- view what individual politicians focus on in their speeches by using the General Network (Speaker) visualization.
- compare differences and similarities of lexical choices of parties and politicians by comparing two general networks.
- observe the change of politicians' and parties' speeches over time by using the year slider.
- see exact values of a node by right clicking on it

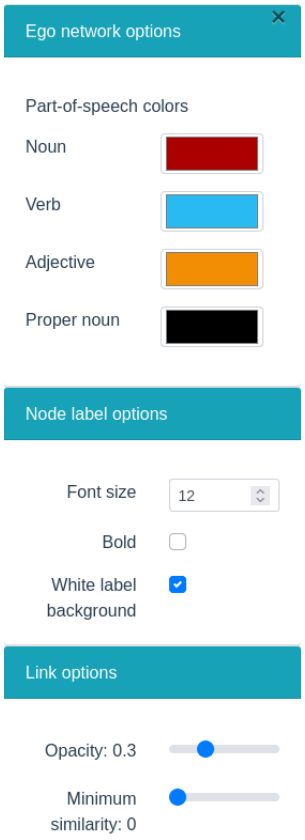

Figure 3.3: Network Graph Visualization Options

3.2.3 Network Graph Visualization Options

In Figure 3.3 options to change the appearance of the network graph visualization can be observed. These options help users better interpret the network according to their preferences. Labels of the nodes are colored according to part-of-speech, and the colors and the font size can be changed. Additionally, the labels can be made bold and can have white background to make them easier to read. Finally edges can be manipulated to be thinner or thicker through their opacity, and they can be shown or hidden according to their minimum similarity, which makes it easier to identify the most similar words or the words used in the same context the most.

3.2.4 Visualization Types

In addition to the graph visualizations of the network types mentioned above, there are also node metric comparison and time series analysis, which provide additional information on the networks.

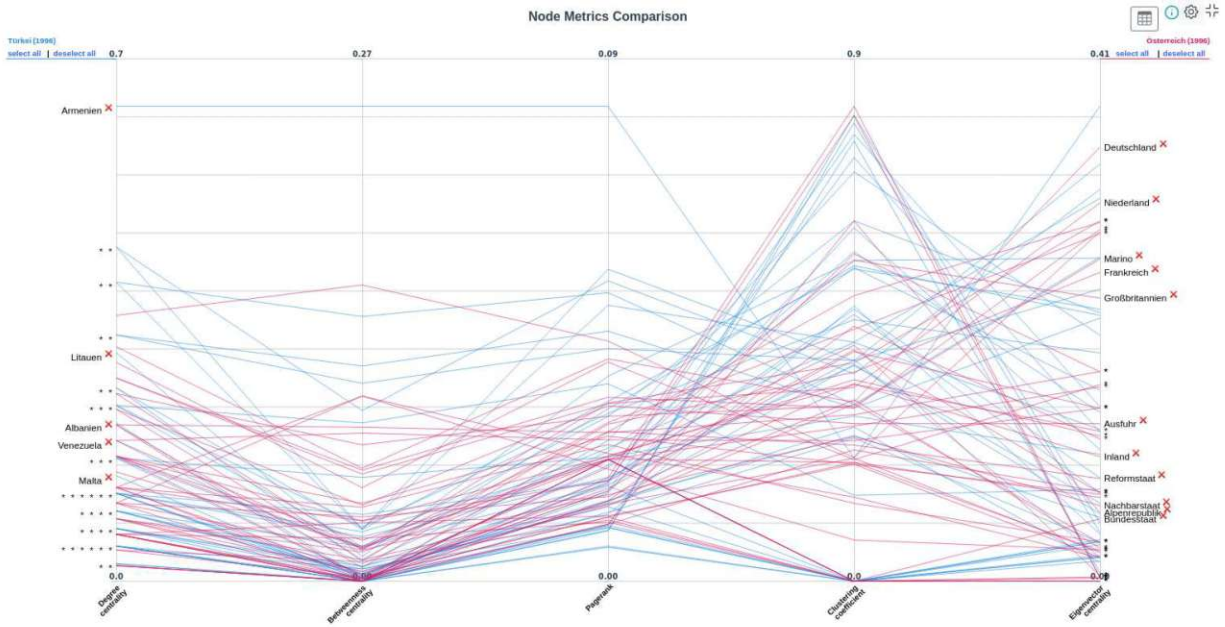

Figure 3.4: Parallel coordinates of the words **Türkei** (meaning Turkey, on the left) and **Österreich** (meaning Austria, on the right) in the year 1996 in all magazines

3.2.5 Node Metrics Comparison

Different metrics can be used to describe the characteristics of nodes and edges within a network graph. The Node Metrics Comparison component helps users to compare different networks based on ten different node metrics.

Node Metrics Comparison has two visualizations: parallel coordinates and table view. They both allow the user to display node metrics of all nodes from up to two different networks. Table view, which is shown in Figure 3.5, is the raw data for the node metrics which are used to draw the graph parallel coordinates, shown in Figure 3.4.

In the parallel coordinates, the x-axis represents the names of different node metrics, while the y-axis shows their values. By default, only five of the possible ten metrics are shown, which can be changed through the “Metrics to Display” section of the settings view. On the top left and right, users can see the name of the network, together with select all and deselect all buttons. Labels for each node can be seen on the left and the right side of the chart, depending on their target word network. All labels have a small x button, which is used to deselect that specific node (remove it from the parallel coordinates). Overlapping labels (words with the same/similar metric value on the first/last axis) are displayed as a *, hovering over the * will expand the view to show the overlapping words. It is very important to note that the network visualization and node metrics comparison components share the same data, which means selecting or deselecting nodes on the network graph will do the same in the parallel coordinates graph and vice-versa. Finally,

Word	Network	Degree Centrality	Closeness Centrality	Betweenness Centrality	Eigenvector Centrality	Pagerank	Load Centrality	Harmonic Centrality	Clustering Coefficient	Normalised Frequency	Absolute Frequency
Albanien	Türkei, 1996	0.225	0.513	0.001	0.193	0.025	0.001	23.333	0.683	7.17e-6	61
Alpenrepublik	Österreich, 1996	0.067	0.366	0	0.054	0.01	0	18.5	0.601	3.76e-6	32
Armenien	Türkei, 1996	0.675	0.741	0.246	0.372	0.085	0.246	33.333	0.206	4.47e-6	38
Aserbaidschan	Türkei, 1996	0.475	0.635	0.027	0.327	0.056	0.027	29.167	0.368	4.70e-6	40
Ausfuhr	Österreich, 1996	0.244	0.445	0.039	0.124	0.031	0.039	23.833	0.263	3.53e-6	30
Ausland	Österreich, 1996	0.067	0.352	0.001	0.07	0.01	0.001	18.117	0.219	1.27e-4	1082
Binnenmarkt	Österreich, 1996	0.089	0.334	0.001	0.026	0.014	0.001	17.5	0.303	1.29e-5	110
Bosnien-Herzegowina	Türkei, 1996	0.075	0.435	0	0.03	0.012	0	19	0.239	8.59e-6	73
Bundesland	Österreich, 1996	0.022	0.022	0	-1.32e-17	0.022	0	1	0	1.11e-4	948
Bundesstaat	Österreich, 1996	0.067	0.356	0.022	0.049	0.011	0.022	18.083	0	7.41e-6	63
Deutschland	Österreich, 1996	0.333	0.467	0.059	0.34	0.04	0.059	25.917	0.299	2.77e-4	2353
Ecu	Österreich, 1996	0.378	0.512	0.154	0.274	0.043	0.154	27.833	0.206	4.00e-6	34
England	Türkei, 1996	0.05	0.449	0.006	0.031	0.01	0.006	19.25	0	8.70e-5	740

Figure 3.5: Node Metrics Table of the words **Türkei** (meaning Turkey, in blue) and **Österreich** (meaning Austria, in red) in the year 1996 in all magazines

users can hover over the lines on the parallel coordinates to see specific values for each metric and word.

The node metrics available for the comparison were calculated using a Python[6] library called NetworkX[7] and they are defined as follows:

- **Degree Centrality:** the total number of edges linked to a node
- **Betweenness Centrality:** the number of the shortest paths that pass through the node; it represents the degree to which nodes stand between each other
- **Load Centrality:** a betweenness-like centrality measure that differs in its definition (uses hypothetical flow process)
- **Closeness Centrality:** indicates how close a node is to all other nodes in the network; nodes with a high closeness score have the shortest distances to all other nodes, i.e. is the most central
- **Harmonic Centrality:** a variant of closeness centrality; higher values indicate higher centrality
- **Eigenvector Centrality:** high eigenvector centrality means that a node is connected to many nodes who themselves have high scores

[6] https://www.python.org/
[7] https://networkx.org/

- **Pagerank:** a variant of eigenvector centrality; the underlying assumption is that a node is as important as the combined importance of the nodes that link to it
- **Clustering Coefficient:** a measure of the degree to which nodes in a graph tend to cluster together
- **Absolute Frequency:** number of occurrences of the word in the selected sub-corpus for the selected year
- **Normalized Frequency:** number of occurrences of the word in the selected sub-corpus for the selected year scaled to 0 and 1 with Min-Max scaler

Users can use the the node metrics comparison visualizations (parallel coordinates and table view) to:

- to see which network has the highest number of semantic neighbors by looking at the highest degree centrality metric.
- to observe to what extent the words in one network differ in frequency by comparing normalized frequency.
- find out how many words are shared by two compared networks by looking at the table view and sorting by word.
- download the table data as CSV/JSON to investigate correlations between centrality metrics.

3.2.6 Node Metrics Comparison Options

In Figure 3.6 table options can be observed. Users can change the number of digits displayed from 0 up to 10. with 0 meaning E Notation. There is also a useful option to show selected words on top of the table, which helps users chose individual words to compare their metrics.

In Figure 3.7 users can select which node metrics should be displayed in the parallel coordinates graph and can change their appearance order.

3.2.7 Time Series Analysis

Time Series Analysis allows you to track changes over time by visualizing the development of (absolute) differences between the values of selected metrics and a specific reference year. Just as the node metrics comparison, time series analysis has two visualizations: time series chart and table view. Table view, which is shown in Figure 3.9, is the raw data for the time series, which are used to draw the graph time series analysis, shown in Figure 3.8.

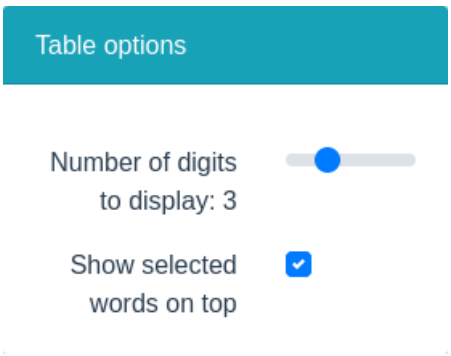

Figure 3.6: Node Metrics Table Options

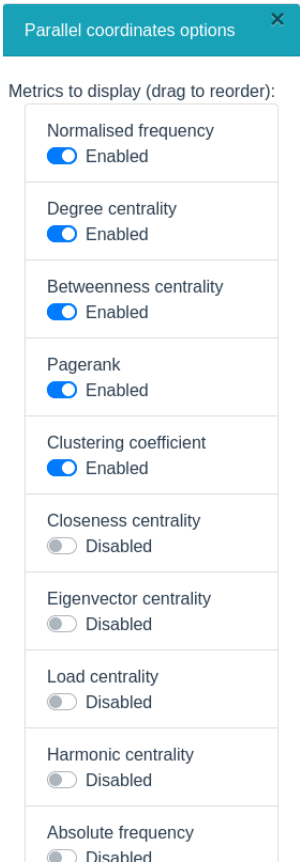

Figure 3.7: Node Metrics Parallel Coordinates Options

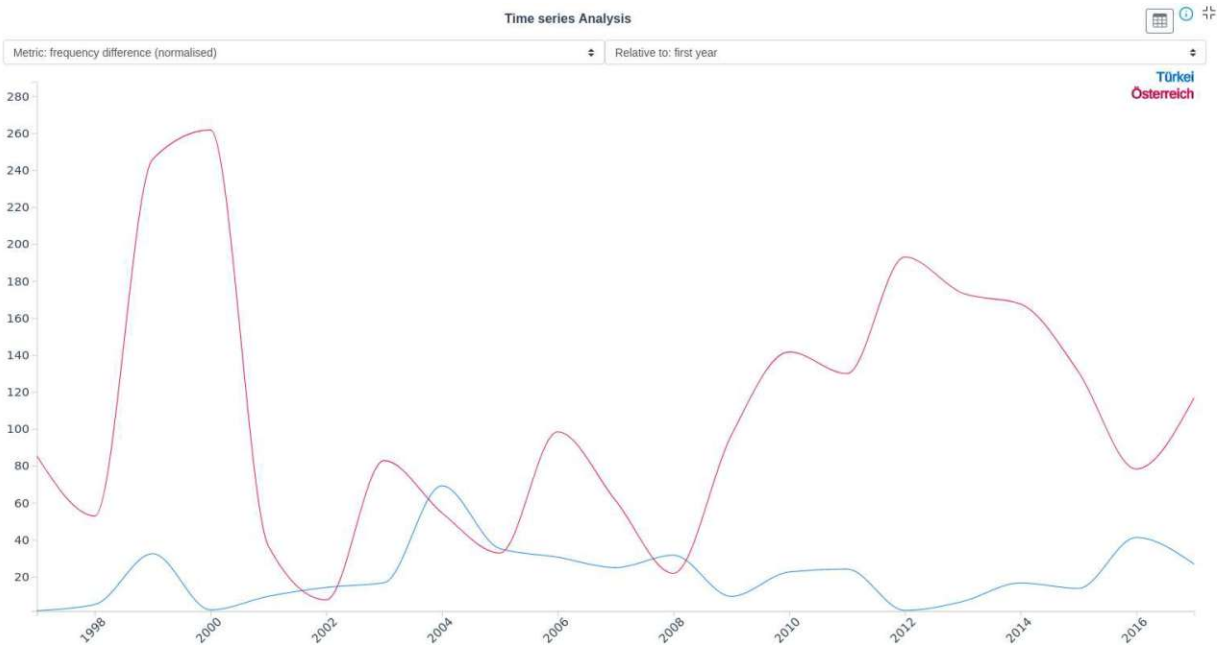

Figure 3.8: Time Series Analysis of the words **Türkei** (meaning Turkey, in blue) and **Österreich** (meaning Austria, in red) in the year 1996 in all magazines

Time series Analysis

Targetword	Metric	Relative to	Year	Value
Türkei	rankDCG similarity	first year	1998	0.23
Österreich	rankDCG similarity	last year	2000	0.253
Türkei	rankDCG similarity	first year	2009	0.285
Türkei	rankDCG similarity	previous year	2010	0.296
Österreich	rankDCG similarity	first year	2008	0.301
Österreich	rankDCG similarity	first year	2012	0.301
Österreich	rankDCG similarity	last year	2014	0.31
Türkei	rankDCG similarity	last year	1997	0.327
Türkei	rankDCG similarity	previous year	2015	0.332
Türkei	rankDCG similarity	first year	2004	0.34
Türkei	rankDCG similarity	previous year	2002	0.343
Österreich	rankDCG similarity	previous year	2015	0.347
Österreich	rankDCG similarity	first year	2017	0.348
Österreich	rankDCG similarity	last year	1996	0.348
Österreich	rankDCG similarity	first year	2009	0.35
Türkei	rankDCG similarity	last year	2013	0.351
Türkei	rankDCG similarity	previous year	2009	0.354
Türkei	rankDCG similarity	first year	2001	0.358

Figure 3.9: Time Series Table of the words **Türkei** (meaning Turkey, in blue) and **Österreich** (meaning Austria, in red) in the year 1996 in all magazines

Users can select the comparison metric they wish from the drop-down menu and select the year you would like to compare the metrics to (first year, last year, previous year). The years are displayed on the x-axis and the y-axis shows the values of the selected metrics.

The metrics available for the comparison are as follows:

- **Frequency Difference:** the absolute difference of normalized frequency values in two years
- **Jaccard similarity:** the similarity between node sets of target word networks in two years by applying Jaccard index [21]. It is defined as the size of the intersection divided by the size of the union of the given sets. The score ranges from 0 to 1.
- **RankDCG similarity:** the similarity between node sets of target word networks in two years measured by applying rank discounted cumulative gain score. Unlike Jaccard similarity, it takes into account semantic closeness of nodes to a target word, and it represents an improved version of the normalized discounted cumulative gain metric described in [22]. The score ranges from 0 to 1.
- **Local neighborhood similarity:** two second-order vectors are created corresponding to two time periods, their length equals the size of nodes union, and cosine similarities between a target word and each node provide scores of the vectors. Then, cosine similarity is computed between these vectors [23]. The score ranges from 0 to 1.
- **Frobenius similarity:** the Frobenius norm [24] of a matrix obtained as the difference between adjacency matrices of target word networks in two years. Adjacency matrices are pre-processed to represent the union of nodes of two time slices. The score does not have a fixed range.

Users can use the the time series analysis visualizations (time series chart and table view) to:

- see if the networks for a target word change over time by choosing the metric jaccard similarity.
- see if the frequency of a target word change over time by choosing the metric frequency difference.

CHAPTER 4

Approach

In the scope of this master's thesis, a semi-automated usability testing framework was developed and implemented as a semi-automated usability testing tool (SAUTO) for the DYLEN tool. The development and usability testing process for the DYLEN tool was used as the case study to test and evaluate the framework and SAUTO. In the following, the case study, the implementations of the DYLEN tool and SAUTO and finally our evaluation approach are described in detail.

4.1 Case Study: Usability Testing

In the scope of this master's thesis, a semi-automated usability testing framework was developed and implemented as a semi-automated usability testing tool (SAUTO) for the DYLEN tool. The development and usability testing process for the DYLEN tool was used as the case study to test and evaluate the framework and SAUTO.

The DYLEN tool was developed with user-centered design principles. Therefore it was planned to carry out formative, iterative and summative usability testing during development. For this work, SAUTO was used during all three types of usability studies, together with manual usability testing in the first two.

During the development of the DYLEN tool, manual usability testing was carried out in two phases, with the first being formative usability study phase and the second being iterative usability study phase. An overview of the phases can be seen in Figure 4.1. In the first phase, after the developers decided that a certain point of minimum functionality was reached, manual usability tests were carried out with five users. The usability testing sessions of these five users were also captured by SAUTO to gather and analyze their usage. The developers then reviewed their findings from manual usability testing and the results SAUTO produced and continued development taking these results into account. Then in the second phase, the same process was repeated: manual usability tests, user

	Formative Phase	Iterative Phase	Summative Phase
Development Duration	~4 months	~1 month	~2 months
User Count	5	6	112
Manual Usability Testing 	✓	✓	✗
Semi-automated Usability Testing 	✓	✓	✓

Figure 4.1: Dylen Tool Development and Testing phases

interaction capture by SAUTO on the same sessions and the reviewing of the results. It must be noted however, that because of technical problems, the SAUTO results in the second phase were incomplete and could not be used as a fully accurate representation of usability issues of the tool. Finally, after the development was finished, the DYLEN tool went officially live and the third phase of the usability testing started, which is called the summative usability study phase, in which there was no manual usability testing conducted. The launch of the tool was disseminated to target groups (researchers and students in the fields of linguistics, political science and journalism) via specific mailing lists, online forums and word of mouth. SAUTO captured 112 sessions with real life usage data, analyzed them and made the results available. We set a target of one hundred sessions for the summative phase and the extra 12 were a welcome addition. The reason for one hundred sessions was to be able to evaluate SAUTO in its intended use, which is conducting usability test with an amount of users that would be impossible with manual usability testing. We aimed to show that the usability testing process could continue without time intensive manual usability testing. Also, we needed a significant sample size difference between the first two phases and the last phase to display the usefulness of having more data for semi-automated usability testing.

Development of SAUTO followed development of the DYLEN tool, because to implement user interaction capture of a feature, the feature had to be fully implemented first. The developer expectations could also be discussed and gathered only after the development of the feature was finished. These facts led to crunch time SAUTO development right after DYLEN tool code freeze until the start of usability tests.

4.1.1 Manual Usability Testing

Manual Usability testing for the DYLEN tool was planned and carried out together with the developers of the DYLEN tool[1]. The three developers worked for the DYLEN project at the Austrian Academy of Sciences. All three of the developers were experienced with usability testing from their previous work including in academics, with one even having some experience in semi-automated usability testing.

In the first round of usability tests five, and in the second round six different users were invited to take part in the usability tests, who were chosen, because they all belonged to one of the target groups of the tool: linguists, political scientists and journalists. According to Turner et al. [25], the magic number of users for conventional usability testing is between 3 and 5. Even though a magic number is hotly debated topic among researchers, we agreed on the number of users based on the research of Turner et al. and our time and resource constraints.

Before the first usability test, a mock-test was conducted with colleagues from the DYLEN team, in order to prepare and train for the real tests. The developers prepared user tasks and they were communicated to the user via a Google Form. The form contained a description of the tool, tasks that user needed to complete and input fields to give feedback on each task and the tool as a whole. First task was "Get familiar with the tool" to be able to allow the user to explore the tool by themselves. The consecutive tasks were concrete tasks with specific goals. The specific tasks with goals were extracted from user stories.

The tests were carried out online using a video conference tool. We decided on online testing in order to prevent the spread of the SARS-CoV-2. Each video call took around one hour in total and the active test of the tool during the video call took around 30 minutes. The users shared their cameras and their screens while testing in order for the observers to observe. The testing sessions were also recorded to be able to view them later to examine user behaviour more thoroughly. Katharina Wünsche BSc. acted as the moderator to help the user get started with the testing process and clarify any questions before the tests. The tests themselves, however were unmoderated and users were asked to use the think-aloud method i.e. say what they are doing as they are doing them. Only in the most extreme cases, where users seemed to be stuck, help was offered.

After the tests were finished, the developers had an internal workshop to discuss their observations from the tests, compare them to the analysis results of SAUTO and decide on further development.

4.2 Implementation

In the scope of this master's thesis, the DYLEN tool web application was used as the subject, for manual and semi-automated usability testing. In order to capture, save

[1] Asil Cetin MSc. (only at the start of the project), Seungbin Yim MSc. and Katharina Wünsche BSc. (last eight months of the project)

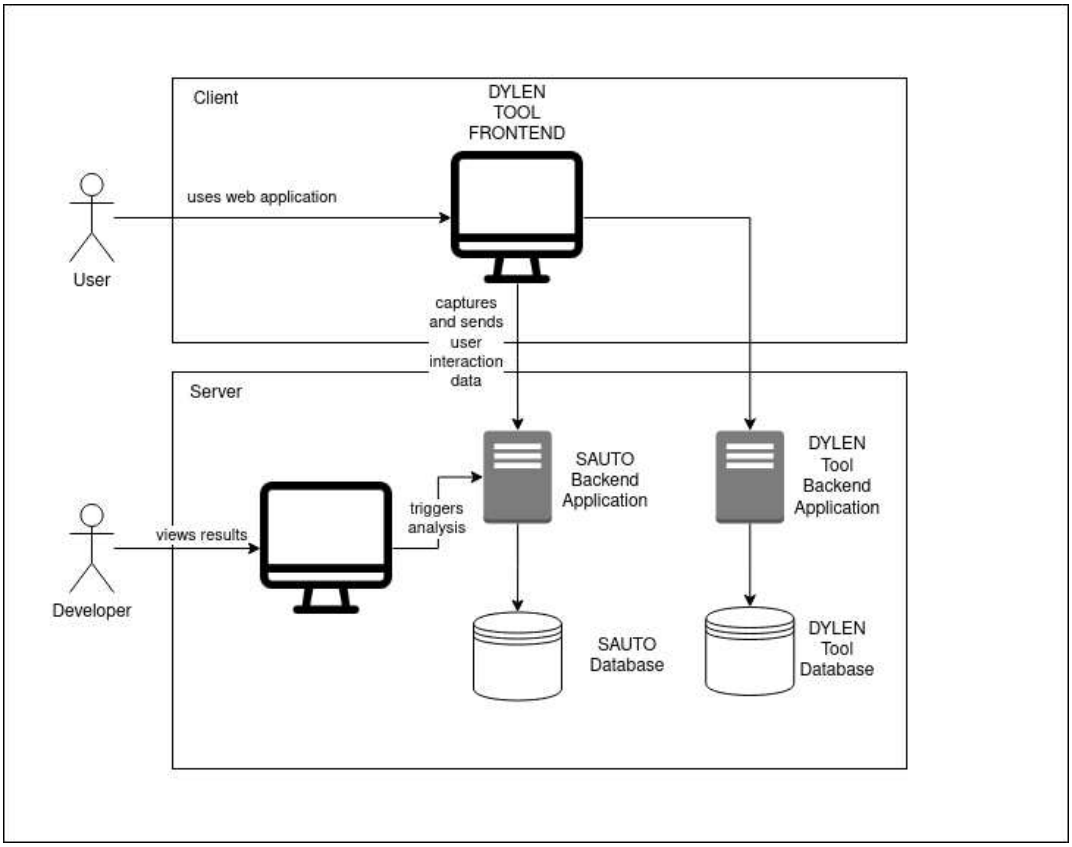

Figure 4.2: SAUTO and DYLEN Tool Architecture

and analyze user interactions during usability testing and after launch of the tool, there needed to be a mechanism to capture the user interactions in the tool and a separate program to analyze the data and display the results. We named this tool Semi-automated Usability Testing Tool (SAUTO). In Figure 4.2 you can see the basic architecture of both tools. The figure also gives an idea about how the two interact with each other and how user data capture, save and analysis processes work.

4.2.1 DYLEN Tool

The DYLEN tool consists of a front-end web application written in JavaScript[2] using VueJS[3] as a framework and a back-end GraphQL[4] application written in Java[5] using the Spring-Boot framework[6]. The back-end application uses MongoDB[7] as its database to

[2] https://developer.mozilla.org/en-US/docs/Web/JavaScript
[3] https://vuejs.org/
[4] https://graphql.org/
[5] https://en.wikipedia.org/wiki/Java_(programming_language)
[6] https://spring.io/projects/spring-boot
[7] https://www.mongodb.com

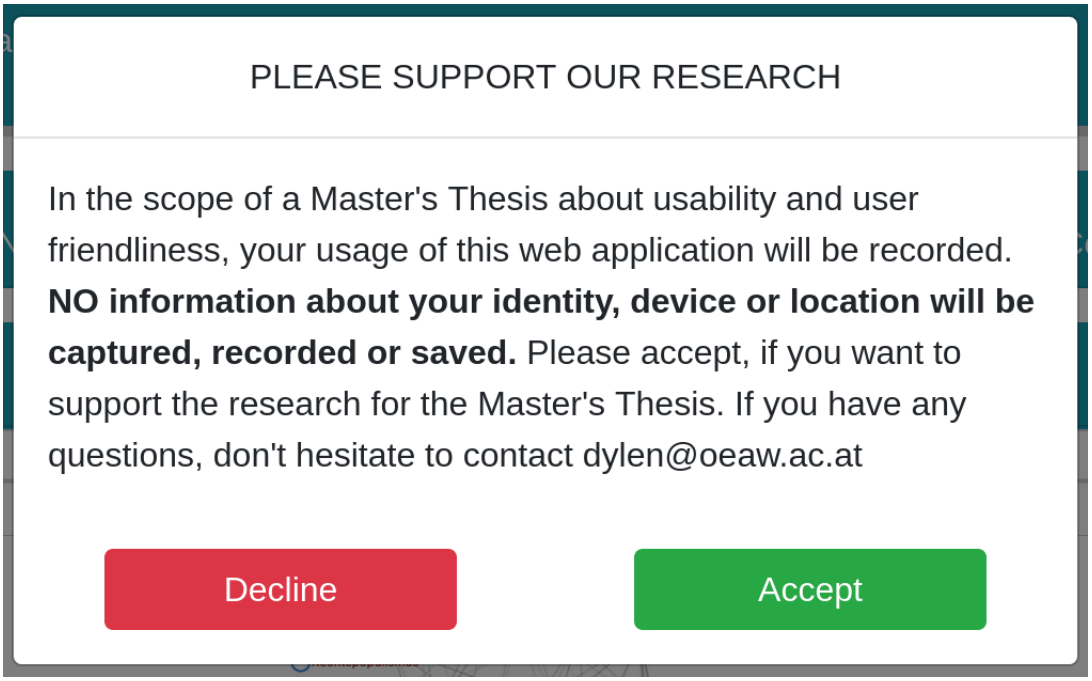

Figure 4.3: Prompt that user sees first upon visiting the DYLEN tool

deliver the network graphs. The relevant part for this master's thesis is the front-end of the DYLEN tool, because it contains the code to capture user interactions. The code to the DYLEN frontend and backend are free to view on Github[8,9].

The first thing the user sees, when opening the DYLEN tool web page, is a prompt asking for permission to capture user interaction data from the session and use it for the purpose of this master's thesis, which is shown in Figure 4.3. Only if the user accepts this prompt, the data will be captured. It is worth noting that the prompt is not about personal data, because for every session, a unique and random id is generated. As it says on the prompt, no information about the user's identity, device or location is captured, recorded or saved.

After the user accepts the prompt, a web socket is opened to SAUTO back-end and all the user's interactions are sent through in real time. User interactions are mostly captured through browser events. Browser events information about the html element, time stamp, and coordinates. This information is used by SAUTO to determine which component on the GUI was interacted with, when it was interacted with and where on the screen it was interacted with. Some components on the front-end don't play nice with browser events and these components need to be extended or modified to be able to capture user events accurately. An example is the simple checkbox. The checkbox

[8] https://github.com/acdh-oeaw/dylen-tool
[9] https://github.com/acdh-oeaw/dylen-backend

registers two mouse clicks on the browser when the user clicks only once and we had to implement a way to ignore one of these to capture accurate data. Additionally, some user interactions are decoupled from browser events such as the resize functionality.

Every user interaction, which can also be called a user event, contains specific data depending on the nature of the event. For example, mouse movement, mouse drag and mouse click events all contain screen coordinates. Coordinates are captured as pixels on the screen, but they are converted to percentages of the width and height of the screen before being saved. This insures that the coordinates are accurate no matter the screen size. Additionally, every user event contains a timestamp to be able analyze interactions chronologically.

The following are all the user events captured:

- Drag
- Key press
- Mouse Click
- Mouse Over (first moment when mouse goes over a component)
- Mouse Position
- Resize (when user rearranges the results graphs or makes them full-screen)
- Scroll
- Timeout (special event when user sends a network query that causes a timeout)

DYLEN tool has a hard-coded version that it sends to SAUTO, which corresponds to the development phase of the tool. All the development until the end of the first round of usability tests was 1.0. Then it was changed to 2.0 until the end of the second round. Finally, with the launch, it was bumped up to 3.0 as its final version.

When we joined the DYLEN project for this master's thesis, only a barebones search query and ego network result was implemented. We took part in the development of the tool together with the developers of the tool. We implemented all the SAUTO specific code, which the DYLEN developers didn't touch.

4.2.2 Semi-automated Usability Testing Tool (SAUTO)

SAUTO is a Kotlin[10] back-end application implemented using the Spring-Boot framework. The code can be found here: `https://github.com/acdh-oeaw/dylen-sauto`. The implementation of SAUTO can be divided into two parts. The first part accepts the connection from DYLEN front-end and saves incoming user interaction data in the

[10] `https://kotlinlang.org/`

MongoDB Database. The second part reads the data from the database, analyzes and delivers a html page containing the results using the Thymeleaf framework[11]. The page also includes the JavaScript library SimpleHeat[12] to display the hotspot maps.

When SAUTO is started, it starts listening on a web socket to accept connections from clients browsing the DYLEN front-end. After the user clicks accept on the prompt, a connection is established through the web socket. SAUTO then creates a session, assigns a random and unique id to it and saves it in the database. Then every user interaction object sent through the web socket is saved into this session in the database. The session object (database entry) contains not only the user interactions but also session start time, end time, duration, exit code and the DYLEN tool version. Start time, end time and duration serve analysis purposes. There is also a cut off at 180 seconds (3 minutes) for a session to be eligible to be analyzed. This insures that short sessions, which don't contribute to usability statistics meaningfully, don't get mixed into results. The limit of 180 seconds was decided by the developers after mock usability tests by determining the optimal minimum amount of time to view the results, interact with and get a basic understanding of them. In order to avoid sessions, in which the user opens the web application and forgets it in a browser tab for a long time, the end time is taken as the last interaction time of the user and not the time the web socket is closed. Exit code is the reason why the web socket connection is closed and serves debugging purposes. Finally, the DYLEN tool version corresponds to the phases of DYLEN development with regards to the usability testing milestones.

After all the sessions are saved into the database, the developer can see the results through the index page of SAUTO. The index page contains a drop-down input with the three version options of 1.0, 2.0 and 3.0. After the user selects one and clicks on analyze button, they will be shown the usability analysis of all the sessions for that version. It must be noted that there are differences of available tables and entries between the versions, because the DYLEN tool was developed in between versions and SAUTO was adapted as well. An example would be the resize count under inferential metrics, because the resize functionality was added with version 2.0. Another difference is the developer expectations, since with each version, the expectations of developers were adapted to the newly implemented features, bug fixes and usability improvements.

In the analysis page, each table and graphic has a description next to it. The contents of this page are discussed thoroughly in the Phases chapter. From the implementation point of view, the results can be separated into per session results and overall results. For example, the developer expectations, which corresponds to the task-based approach, evaluates each session separately, while the other approaches all evaluate by aggregating all the sessions together. The overall results have different types of analysis. For example, click count on a component is aggregated and divided by session count to get the average click count per session. Another type uses the timestamps to sort events chronologically to find out repeated patterns.

[11] https://www.thymeleaf.org/

[12] https://github.com/mourner/simpleheat

The features of the DYLEN tool were based on user stories defined during the project planning process. User stories were defined as who a user is and what do they want to accomplish with the tool. In the formative phase of the testing, there were 9 unique user stories. With more features added in the iterative, there were 11 and finally for the summative phase there were 19 unique user stories. These user stories were implemented into SAUTO as completion events. A completion event was defined for each user story and these completion events make up a significant portion of the analysis. For example, one user story that should be completed roughly at the start of each session is *"ego network word search"*. The user selects a corpus, a sub-corpus and a target word, then clicks on the visualize button to see the results. The clicking of the visualize button indicates that the user story *"ego network word search"* is completed and it can be registered as a completion event for the story.

Description of SAUTO Results

The analysis page of SAUTO[13] contains the results of the usages of the selected version of the DYLEN tool. In the following list, all of the different results in the tables, lists and hotstpot maps are explained. The list is divided into the four approaches to semi-automated usability testing: metric based, pattern matching, task based, inferential. Additionally, there is the general results section, which are an aggregated and ordered combination of the results from all the approaches. The ideas for tables, lists and hotspot maps are mostly from the research described in the chapter State of the Art with a few being original/modified versions.

It must be noted that not all results are available on all versions, for example, clicks per page is only available in version 3, because pages were introduced with version 3 of the tool.

- **GENERAL RESULTS**
  - **Best/Worst Performing Elements:** Two lists of the top three best/worst performing elements in the UI of the web application. The lists are compiled through aggregating the results from all the following analysis methods and extracting the best and worst performing elements. The different methods are weighted when calculating the best and worst. For example, user story completion count is more important than user story completion interaction. The items in the best list are generally intuitive and easy to use for users. And items in the worst list imply the opposite. Developers should look into the elements and try to change their design and functionality to fix possible usability problems.
- **METRIC BASED RESULTS**

[13] https://sauto.acdh-dev.oeaw.ac.at/index.html

- **Metrics Table:** This table shows some general statistics. These stats can be compared to other versions of the web application.
- **User Story Completion Count Per Session:** This table shows how many times a user story is completed in one session. The user story with the highest result is the most commonly completed user story. Developers should examine all the user stories with a count of less than 1, which means not all users were able to complete the user story. If there is a story with a count of near or equal to 0, that user story points to a usability problem.
- **User Story Completion Time Statistics (sorted by Average Duration):** This table shows how long it took to complete each user story. The time is calculated as the following: the time that passes since last completion event until this user story's completion event counts as the duration (first one starts from session start). The user story with the highest average duration is the most problematic one.
- **User Story Completion Interaction Statistics (sorted by Average Count):** This table shows how many user actions are needed to complete each user story. User actions are anything from mouse movement to clicks and every interaction that SAUTO captures. The one with the highest average is the most problematic one.
- **First Click Duration (from session start) for each element:** This table shows the time it takes users to click on the element for the first time in a session. It gives an idea of how easily detectable the elements are. It should be kept in mind however that some elements are not visible at the start of the session and come into view later. It also gives a good overview of the workflow of the user (sequence of interactions in order).

- **PATTERN MATCHING RESULTS**
  - **Mouse Click Repetitions:** This table shows repeated mouse clicks on the same item throughout all sessions (minimum five occurrences). The more an item is clicked repeatedly, the more it might point to a usability problem. It is up to the developer to decide if repetition on a specific item is bad.
  - **Repeated Patterns x2:** This table shows pairs of clicks, that are repeated all sessions (min 5). Repeated click patterns means that an element often follows the other in user click order. Depending on developer's expectations, repeated patterns can provide usability problems or can confirm expected user action order.
  - **Repeated Patterns x3:** This table shows triples of clicks, that are repeated all sessions (min 5). Repeated click patterns means that an element often follows the other in user click order. Depending on developer's expectations, repeated patterns can provide usability problems or can confirm expected user action order.

  - **Repeated Patterns x4:** This table shows quadruples of clicks, that are repeated all sessions (min 5). Repeated click patterns means that an element often follows the other in user click order. Depending on developer's expectations, repeated patterns can provide usability problems or can confirm expected user action order. We decided against including repeated patterns more than 4, because firstly we didn't have enough user interactions in the first two phases (only 5 and 6 users generated a low sample of user interaction data), secondly it became clear that the repeated patterns more than 4 were providing the same information as mouse click repetitions.
  - **Repeated User Story Patterns x2:** This table shows pairs of user stories completions, that are repeated all sessions (min 5). Repeated user story patterns mean that a user story often follows the other in user interaction order. Depending on developer's expectations, repeated user story patterns can provide usability problems or can confirm expected user action order.
  - **Repeated User Story Patterns x3:** This table shows triples of user stories completions, that are repeated all sessions (min 5). Repeated user story patterns mean that a user story often follows the other in user interaction order. Depending on developer's expectations, repeated user story patterns can provide usability problems or can confirm expected user action order.
  - **Repeated User Story Patterns x4:** This table shows quadruples of user stories completions, that are repeated all sessions (min 5). Repeated user story patterns mean that a user story often follows the other in user interaction order. Depending on developer's expectations, repeated user story patterns can provide usability problems or can confirm expected user action order. For the same reason as repeated patterns, we stopped at x4.

- **TASK BASED RESULTS**

  - **Developer Expectations:** This table shows a list of developer expectations, that were determined by the developers before the tests and how many sessions that these expectations were fulfilled or not. This table is very useful for the developers to see if their application is being used as they intended.

- **INFERENTIAL RESULTS**

  - **Mouse Click Hotspot Map:** This hotspot map shows where the users click the most. It is helpful for visualization of users' mouse clicks which can also be seen in the table below. User can change screenshots to see clicks on each page/screen.
  - **Click Counts Per Session For Items With More Than 1 Click:** This table shows click counts for each element with more than 1 click in average. It helps with the visualization of the hotspot map above.
  - **Mouse Movement Hotspot Map:** This hotspot map shows mouse movements. It is helpful for visualization of users' mouse movements which can

also be seen in the table below. User can change screenshots to see mouse movenets on each page/screen.

- **Mouse Over Count Per Session:** This table shows the mouse over counts, which means when the user brings the mouse onto the element from the outside. It helps with the visualization of the hotspot map above.
- **Mouse Drag Map:** This map shows where users use drag and drop. Network graphs contain drag and drop functionality and it is useful to see how it is used.
- **Durations Per Page:** This table shows how long the users spend on each page. It should be noted that the user starts at ego network page. The more the user spends time on a page, the more that page offers. This is a metric to show us which page is more interesting and more usable.
- **Clicks Per Page:** This table show the clicks per page average. More clicks per page means users interact more with that page. This table provides us insights into which page out of the 3 provide more interactions and therefore are more usable. It could also show discrepancies between the pages in terms of possible interactions.
- **More Inferential Metrics:** This table shows more inferential metrics. First one is key press counts and the second one is scroll counts. Key presses are used in the target word search and scrolls are used to zoom in and out of the graphs. These metrics are useful to compare between different versions of the web application.

4.3 Evaluation

SAUTO was evaluated with two approaches within the DYLEN project. The first approach was quantitative, in which the results of SAUTO analysis were compared to results from manual usability tests. This is what the studies into semi-automated usability testing did to evaluate their proposed tools [20, 7, 5, 9]. The second approach to evaluate was qualitative: interviews. We included qualitative interviews in this thesis, because the semi-automated usability results from the summative phase did not have manual usability results to compare to. So there was a need for a different approach to evaluate SAUTO, especially the results from the summative phase. There were three interviews conducted: two with developers of the DYLEN tool, who had first hand experience with SAUTO during development and one with an expert on usability, who took no part in the development. The reason for the third interview was to get an unbiased analysis of SAUTO by an external party.

In the first evaluation approach, we compared numbers of usability problems discovered plus usability expectations confirmed (correct usages). Because SAUTO results not only display usability problems but also give information about correct usages of the tool as the developers intended. In the formative and iterative phases of the usability testing,

the goal of SAUTO was to discover the similar number of usability problems compared to the manual usability testing. In the summative phase, the goal was to discover similar number of or more usability problems to the first two phases. The reason of expecting more from the summative phase is, because it contains significantly more sessions and therefore has much more usability data to analyze. This is the strength of semi-automated usability testing, which enables to test with more users with less resources.

In the second evaluation approach, we conducted interviews with the two developers of the DYLEN tool and an external usability expert following the guide by Daniel et al. [26]. This guide defines three categories of interviews, of which standardized open-ended interviews was chosen. These kind of interviews are extremely structured in terms of wording of questions. Participants are asked identical questions but the questions are worded to be open ended. This allows participants to contribute as much detailed information as they wish. We employed this interview approach in this work with some adaptations. The interviews with the developers were conducted after the last workshop, in which we discussed the final results of the summative phase. The interview with the external expert was preceded by an in depth presentation of DYLEN, SAUTO, the development process of both and the results from all three phases. The questions were identical for all participants and were all open ended. Following the guide of Daniel et al. we selected the participants with two criteria: having experience with SAUTO (two DYLEN developers) and having experience with usability testing in general (external usability expert). We then created research questions by keeping the wording open-ended, clear and neutral. We added phrases like "why or why not" or "please elaborate" to get the maximum amount of information from the interviewees. The two major changes to the guide of Daniel et al. were that the interviews were done in written form utilizing Google Forms and that there were no follow-up questions asked. The reason for using online forms was to give the interviewees enough time to think and formulate their answers. They filled the form in their own time and had easy access to the SAUTO results if they wished to take a look. After all the answers were gathered, we did not find any reason to ask follow-up questions, because the answers given by the interviewees were clear enough. When interpreting the answers, Daniel et al. describe that researchers must make sense out of what was uncovered and compile the gathered data into sections or groups of information, also known as themes or codes. These themes or codes are consistent phrases, expressions or ideas that were common among the interviewees. In this work, we compiled the answers into themes, which are discussed in the Evaluation section of the chapter Evaluation.

CHAPTER 5

Phases

In this chapter the results of all three phases are discussed. In order to keep the chapter from becoming too long, some tables and figures are located in the Appendix A B C.

5.1 Formative Phase

The Formative phase was the first round of usability tests for the DYLEN tool. It is called formative because it was the first time real users tried the tool, so the first usage examples were formed. Below are the results for this phase with each list, table and figure explained.

5.1.1 General Results

Table 5.1 shows all the user stories that the developers prepared. The user stories correspond to the features of the tool, which are described in the chapter DYLEN. The element id corresponds to the component's id in the tool. Clicking on the component with the id confirms the user story as completed. The following results and discussions all make use of this user story to element id relationship. It is important to note that some elements correspond to the same story. For example, to complete the story *"Visualize a different year"*, users can either do it by clicking on "year-slider-pane1" or "year-slider-pane2"

Table 5.1: Formative Phase: User Stories to Element Ids

User Story	**Element Id**
Basic Word Search (Visualize a word)	queryButton-pane1
Visualization (Select a word in the graph)	pane1-node
Visualization (Select a word in the graph)	pane2-node
Graph Comparison	queryButton-pane2
Table View Visualization	table-button
Select Table Item	table-item
Visualize a different year	year-slider-pane1
Visualize a different year	year-slider-pane2
Sort Table	table-sort
Remove line from Parallel Coordinates	x-button-parallel-coordinates
Settings Button	settings-button

The best and the worst performing elements do not require much interpretation. As explained in the section Description of SAUTO Results, they are the result of aggregating all the analysis methods. Analysing the rest of the results confirms the best and worst performing elements. After the tests, the best performing elements were changed as little as possible, whereas the worst performing elements were redesigned according to these results.

Best Performing Elements:

1. year-slider-pane1
2. pane1-node
3. table-sort

Worst Performing Elements:

1. year-slider-pane2
2. x-button-parallel-coordinates
3. selectSubCorpus-option

5.1.2 Metric Based

Table 5.2 shows some general metrics about all the sessions in the phase. There were five users and six sessions, because one user closed the tool in the middle of the test. Afterwards, they started from the beginning, so this event served to gather two sessions, with one being from a new user and one being from a more experienced user. Possible unique user story count is self explanatory, as in, it is the number of user stories but

not the number of corresponding element ids. Finally, the last three metrics don't mean much by themselves but are a good indicators of usability changes when compared to other phases, so they will be more relevant in the iterative and summative phases.

Table 5.2: Formative Phase: General Metrics

Metric	**Result**
Eligible Session Count	6
Possible Unique User Story Count	9
User Story Completed Per Session	38.33
Average User Story Completion Duration (s)	21.34
Average Session Duration	30m2s

The top four user stories in Table A.1 are *"Visualize a different year"* on the first pane, *"Visualization (Select a word in the graph)"* on the first pane, *"Sort Table"* and *"Select Table Item"*, and their occurrence count suggest that they are intuitive and used multiple times per session. However, *"Visualize a different year"* (year-slider-pane2) was not completed even once, so the second graph year slider is not visible/intuitive to the users. Remove line from Parallel Coordinates (x-button-parallel-coordinates) is also rarely used and should be looked into as a possible usability problem.

In Table A.2 *"Select table item"* and *"Visualize a different year"* on the first pane both seem to be intuitive enough for the users, because they seem to complete these tasks quickly. *"Visualization (Select a word in the graph)"* for the second pane, which is the result of the second search (pane2-node) was not used much in the previous table A.1, so it being high in this table doesn't mean that is was user friendly, because according to the previous table, most users completed the story once or not at all, so having the one user complete it fast doesn't mean it is user friendly. Same goes for the *"Remove line from Parallel Coordinates"* story. The stories with the longest time are definitely *"Basic Word Search (Visualize a word)"* on the first search form and *"Graph Comparison"* on the second search form, because they require more actions to complete (filling search form elements), so it is to be expected.

In Table A.3 similar to the previous table, we can ignore user stories that have a small data sample: *"Remove line from Parallel Coordinates"* and *"Visualize a different year"* for the second pane. *"Select Table Item"* and *"Sort Table"* in this table show that once the user starts interacting with the table, they don't change their focus away, which is good. Same can be said for both graphs because of the story *"Visualization (Select a word in the graph)"* on both panes. However, there is a difference of three interactions on average between the panes. This can be the result of low sample data for pane2-node. Similar to the previous table, *"Basic Word Search (Visualize a word)"* on the first search form and *"Graph Comparison"* on the second search form require the most interactions, which works as expected as users need to interact with the search form.

Table A.4 tells us duration it took the users from the start of the session to discover the elements. It also gives us an idea about the order of the user flow. The second item on the list is info, which is the first of two elements that the user sees. First items to be discovered and used (other than info) are the first search form options, which makes sense, because they are the other element group that the users see at first. Then the user starts playing around with the graph with zoom-in-button-pane1 and pane1-node. The user also discovers the second query button. However, they still interact with the first graph more. Settings button is also discovered after the first query and also interacted with around this time. However the user doesn't interact with specific options and goes back to the results. The last items that users discover are parallel-coordinates and table views. Finally, the last thing the users do is playing around with the settings. It is worth noting in this table that the form fields and the graph for the second query are all discovered in the end. This shows that the comparison functionality is not very intuitive to the users.

5.1.3 Pattern Matching

In Table A.5 we can see that most repeated clicks happen inside the info element. Info is the first thing that the user sees and only contains non-interactive image and text about the tool. Repeated clicks means users are either trying to click the images (which are screenshots) or highlighting the text, which means the text is too long to read. The elements year-slider-pane1, table-item, table-sort and pane1-node are the three next elements (except selectTargetword-pane1) that cause repetitions which work as expected, because those items need to be played around with. For example the user can click on one year and click on another right after to compare. Select form dropdown elements are the next items in the table. SAUTO differentiates between the select target word dropdown and its options. So this means users click on the dropdown menu items repeatedly. This points out to the possibility that the form might be somewhat buggy/unresponsive.

In Table A.6 most elements of the table point to a correct order of usage of form elements. "selectTargetWord-option -> queryButton-pane1" suggests that the users skip selectYear dropdown menu and use the year slider instead. This happens only for the first graph, as the users never interacted with the second graph's year slider. "year-slider-pane1 -> table-button" and "table-button -> year-slider-pane1" point to a hidden relationship. This means the users want to compare different years and visualize both table view and parallel-coordinates view. To make this comparison easier, a possible solution could be to show both components side by side. "select-all-checkbox-pane1 -> pane1-node" and "table-sort -> table-item' are expected repetitions.

In Table A.7 first repeated item contains selectTargetword-pane1 twice, which means it is clicked twice before selecting a target word option. This points to a usability problem. User clicks twice where one should be enough. This suggests that the form is buggy/unresponsive. Rest of the elements in the table suggest correct usage of the form and tables. For example the second item on the table is the correct order of completing the user story *"Basic Word Search (Visualize a word)"*.

Table A.8 same as the previous table, shows the problem with selectTargetWord input in the form, because of the unexpected repetition.

In Table A.9 the orders of user stories are as expected. The same exact year-slider-pane1 to table-button relationship from the table A.6 can be seen here.

In Table A.10 table items are used in succession to one another as expected.

In Table A.11, same as the previous table, table items are used in succession of one another.

5.1.4 Task Based

In Table A.12 we can observe the specific expectations of the developers compared to what real users actually did. We go over each expectation specifically below.

Analysis of Developer Expectations

1. Settings are opened but not interacted with much as expected, which means the default values are good or the settings were not interesting to the users.
2. First user event to be completed is as expected *"Basic Word Search (Visualize a word)"*.
3. More than half used the zoom in the first network graph.
4. Less than half used the zoom in the second network graph.
5. Most users used the search form in the correct order.
6. Select all checkbox is used as expected. However, this is the first and only false positive result of SAUTO, because we were able to observe users struggle with these checkboxes during the manual usability testing. Although the tests were unmoderated, we had to intervene and point the checkboxes to the users in order to continue. So SAUTO correctly points out that they were clicked on but we as the informed observer know that this expectation was not actually fulfilled.
7. This has the same conclusion as the item 6 on this list, because it is the other checkbox. So this is the same false positive.
8. This has the same conclusion as the item 1 on this list. Settings are opened but not interacted with much.
9. Most users played around with the network graph as expected.
10. It is again evident that some users didn't try the second pane.
11. All users searched for a word as expected.

12. One session, most probably belonging to the divided one, didn't query for the second graph, which is fine.
13. Half the users didn't bother with the settings, which is as expected.
14. All users opened the table view as expected.
15. Half the users didn't interact with specific elements in the table, which should be looked into.
16. Table sort was used by most users.
17. x-button-parallel-coordinates performed badly as users mostly didn't find this functionality
18. Year slider in the first pane was used substantially to change years.
19. Year slider in the second pane was not used at all, even though the first one was used. This is a big usability problem.

5.1.5 Inferential

From Figures A.1 A.2 A.3 A.4, which depict mouse click hotspots, we can infer some information. The forms are used the most as expected. The other elements on the screen are also used enough. There is one hotspot to the middle right, which can't be explained at a first glance. There is no item there at all but there are somehow multiple clicks. This is caused by bad responsive design of the app. On smaller screens parallel-coordinates metric labels were overlapping, which lead users to highlight the text. This highlight action was the cause of these phantom clicks.

In Table A.13 form elements are clicked the most as expected. Info has a high click count and this might be a problem, which was explained in mouse click repetitions.

In Figure A.5 we can see that there is mouse movement everywhere on the screen, where there is an element. A particular complaint of one tester during the manual usability tests was that the top left DYLEN title is not clickable. Most users haven't hovered over it thinking it as a button, which works as intended. There is no homepage to go to, so the title is not a button. This is a good example of a true negative and an example of how semi-automated usability tests can sometimes deliver results, which manual usability tests might miss.

In Table A.14 we can see the users mostly go between the form on the left and the results on the right. Users regularly switch between the table and the graph view of the parallel-coordinates.

In Figure A.6 we can observe that users seem to drag the nodes outwards from the center. This might mean that the starting positions of the nodes are too close to each other and a larger graph might be necessary to view all nodes efficiently.

The metrics in Table A.15 don't mean much by themselves but they will be useful to compare between the next iteration of the web application and this one.

5.1.6 Summary

In Tables 5.3 and 5.4, we can see what usability problems and bugs were discovered by manual usability testing and by SAUTO. Manual tests were able to discover 19 problems, while SAUTO was able to discover 10, with only 3 of them not being discovered in the manual testing. In Table 5.5 we can also observe the 9 cases of correct usability discovered, as in, the users used the tool as intended by the developers. It is possible to also observe this in manual usability testings but observers are more inclined to focus only on the problems.

Table 5.3: Formative Phase: Manual Usability Test Results

Description	**Category**
Visualizations need to be larger, zoom button needed	Usability
Connection between EgoNetwork and Parallel Coordinates Visualization not clear	Usability
Query buttons sometimes update both panes	Bug
Select year dropdown menu not used at all in some cases	Usability
Blue and black text labels are hard to differentiate	Usability
More info on Ego Network title needed (PoS, full newspaper name)	Usability
Nodes are not clickable, because they are overlapping	Usability
Users had problems with x-button-parallel-coordinates	Usability
Select all button not used intuitively	Usability
Labels are needed on left and right side of the parallel coordinates, and also line colors need to be labeled	Usability
When deselecting nodes on parallel coordinates, the whole visualization is removed from screen	Bug
Users don't know how to interpret metrics on parallel coordinates	Usability
Users don't know how to interpret metrics on parallel coordinates	Bug
Select all checkbox doesn't work correctly	Bug
Auto complete doesn't work correctly	Bug
Table view first two columns should be fixed	Usability
Metrics on parallel coordinates look bad on smaller screens	Bug
It is counter-intuitive that the second graph is under the first one (users dont use second one as much)	Usability
Table view needs a filter for only selected ones	Usability

Table 5.4: Formative Phase: SAUTO Usability Test Results - Problems

Description	**Category**	**Problem found in**	**Found also in Manual**
year-slider-pane2 not used at all	Usability	Metric based, Task based	Yes
x-button-parallel-coordinates rarely used	Usability	Metric based, Task based	Yes
Second network graph is used much less	Usability	Metric based, Task based	Yes
Too many clicks on info (either too much text or users trying to click non interactive images)	Usability	Pattern matching, Inferential	No
select target word has too many repeated clicks	Bug	Pattern matching	Yes
select corpus, select sub corpus and select year drop downs have too many repeated clicks	Bug	Pattern matching	No
“selectTargetWord-option -> queryButton-pane1” suggests that the users are skipping selectYear dropdown menu and use the year slider instead (only for the first graph)	Usability	Pattern matching	Yes
“year-slider-pane1 -> table-button” and “table-button -> year-slider-pane1” point to a hidden relationship. This might mean the users want to compare different years and visualize both table view and parallel-coordinates view. Maybe the users want to see both views side by side.	Usability	Pattern matching	No
Parallel-Coordinates metrics look bad on smaller screens, and cause clicks to highlight to be able to read	Bug	Inferential	Yes
The starting positions of the nodes are too close to each other, which causes many drags from center outwards, possible solution is to make visualization bigger	Usability	Inferential	Yes

Table 5.5: Formative Phase: SAUTO Usability Test Results - Correct Usages

Description	**Problem found in**
Year slider is used by most users and is intuitive	Metric based, Pattern matching, task based
Table view is used by most users and is intuitive	Metric based, Pattern matching, task based
Network graph is interacted with by most users	Metric based, Pattern matching, task based
General workflow order of the users match developer expectations	Metric based
Users don't interact with settings much	Task based
First user story completed is visualization as expected by the developers	Task based
Search form was used in the correct order	Task based
Select all checkbox was used by most users (discrepancy with manual usability tests)	Task based
DYLEN logo is not clicked as expected	Inferential

5.2 Iterative Phase

The Iterative phase was the second round of usability tests for the DYLEN Tool. It is called iterative, because it comes after the formative and it iterates to get a second look on the tool.

Unfortunately, there was a bug in the tool during the usability tests, which caused half of the data captured of each session being lost. As explained in the section Manual Usability Testing, a Google Form was used to gather data from the participants during tests. Users start by reading the information on the form, then clicking on the link to the DYLEN tool to get started. After each task they finished, they would go back to the form to answer questions about the task. The Nginx webserver that hosts SAUTO would however cut any websocket connection to the tool after inactivity of 60 seconds. If the users spent more time than one minute answering the questions in the form, the connection would be cut and the rest of the session would not be captured. This bug did not affect the formative phase by some miracle (each session duration in SAUTO was compared to videos of the manual usability tests) but it unfortunately cut all of the sessions in the iterative phase by half.

The first task that the users completed before the sessions being cut off was the explorative task, in which the users try to get a feeling of the tool, try to understand the components and click around. This unfortunate bug renders some of the results obsolete. Below are the results that provide some information despite the unfortunate bug. General Results and any result with a discrete count is obsolete but other results might provide some information, so the obsolete results are omitted below. They are however available on the SAUTO page.

After the formative phase, the developers worked on new features, usability improvements and bug fixes. The results from both manual usability tests and SAUTO were discussed inside the development team. After the discussion, the problems were prioritised and fixed until the tests for the iterative phase started. The following are the main changes between the phases:

- Select year dropdown menu was removed
- Year slider was moved to the top of the ego network from the bottom
- Time series analysis was introduced
- Select all checkbox was moved to top right from top left
- Zoom in/out buttons were moved to bottom right from top left
- Reset Zoom button was introduced next to zoom in/out buttons
- Query button was renamed to Visualize

- Reset Query button was introduced next to Visualize
- On either side of parallel coorinates, name of the targetword with its year was introduced
- On either side of parallel coordinates, select/deselect all buttons were introduced
- Resizing result by dragging the separator line functionality was introduced
- Fullscreen results functionality was introduced
- Info page was fleshed out to have more information
- Info page was separated into tabs to add more structure
- Info button was introduced to allow users to go back to the info page
- Settings button was changed to a settings icon from text
- General bug fixes

5.2.1 General Results

In Table 5.6 the updated user stories with their relevant element ids can be viewed. The additions are *"View Time series"* and *"Reset query for network 2"*. The reason for reset query being a user story only for the second network is that the manual usability tests had instructions to change the second pane explicitly. So the developers added resetting it as a required user story.

Table 5.6: Iterative Phase: User Stories to Element Ids

User Story	**Element Id**
Basic Word Search (Visualize a word)	queryButton-pane1
Visualization (Select a word in the graph)	pane1-node
Visualization (Select a word in the graph)	pane2-node
Graph Comparison	queryButton-pane2
Table View Visualization	table-button
Select Table Item	table-item
Visualize a different year	year-slider-pane1
Visualize a different year	year-slider-pane2
Sort Table	table-sort
Remove line from Parallel Coordinates	parallel-coordinates-x-button
Settings Button	settings-button
Compare Time Series	timeSeries-relative-option
Reset query for network 2	resetQueryButton-pane2

5.2.2 Metric Based

User Story Completion Count/Time/Interaction tables under the metric based approach are unfortunately obsolete because of the bug. They contain discrete numbers and totals per session, which don't make sense if the full session was not recorded. Only the two most basic metrics are valid and can be seen in Table 5.7.

Table 5.7: Iterative Phase: General Metrics

Metric	**Result**
Eligible Session Count	6
Possible Unique User Story Count	11

There is also Table B.1, which can give us some details, because this table mostly relies upon the first part of the sessions and not the lost second parts.

The first three items on the list are all in the info page, which is the first page the user sees, so it is the obvious logical conclusion that they should encounter and discover them first. The next six elements are all the search form options and the visualize button in the search from. This behaviour is the same as in the formative phase. However it took users roughly three times longer in this phase to discover the search form. This is to be expected as there is much more information on the info page. Next on the list is the year slider of the first pane. Although it is high on the list, the manual usability tests found out that the users still struggle with discovering it, so this can be considered another false positive for SAUTO. After that, the list becomes very mixed, because the discovered elements are all in different components. There is pane1-node, settings options, time series options all mixed in chronological order. This might mean that in the explorative phase, the users don't pay much attention to a single component but divide their attention on multiple components. This leads us to two possible conclusions: there are too many components at once on the screen for a first time user and each component is not self explanatory enough to hold the users' attention. It is also worth noting that the second query is discovered in the later stages of the exploratory session, which means it is not obvious enough. This is the same behaviour as the formative phase. The table button in the node metrics comparison is also discovered very late. The last element is the same exact element as the formative phase: a specific option in the settings tab, but this works as intended, because options are designed for power users and not first timers. Finally, the duration averages of discovering elements is longer than the formative phase, which makes sense, since there are more elements to be discovered, each of which holds the attention of the user for a period of time.

5.2.3 Pattern Matching

In Table B.2 we can see expected repeated clicks for the first four elements: pane1-node, zoom-in-button-pane1, table-item and selectTargetword-pane1. Node and the zoom in button are both parts of the ego network graph, which require repeated clicks to make

use of. Table item can also be repeatedly clicked because the user is clicking on the checkboxes on the table to select or deselect items. Select Target word can also be double clicked to highlight all the text, so repeated clicks are to be expected. info-tab-content however is the same problem as the formative phase: users are clicking on non-interactive images or highlighting text, which is too long. Rest of the items in the list don't have many repetitions and the repetitions on those elements are acceptable and in some cases to be expected.

In Table B.3 most elements of the table point to a correct order of usage of form elements. "queryButton-pane1 -> selectTargetword-pane1" might point to a problem. But since it happened only 5 times, it can be written off as users searching for the wrong word and trying to fix their mistakes by directly searching the correct word.

In both Tables B.4 and B.5, the patterns seem to be nothing out of the ordinary. The only element in the latter table might suggest a repeated click problem, but as explained before, users might be double clicking to highlight the text and its count is only five, so it is not a sign of a significant problem.

In both Tables B.6 and B.7, point to a expected flow of the visualize button leading to the year slider and then to the nodes. The table with four repeated user story patterns is empty, so it is omitted.

5.2.4 Task Based

The numbers in Table B.8 are unfortunately also obsolete but we included the table in order to list the developer expectations for this phase.

5.2.5 Inferential

From Figures B.1 B.2 B.3 B.4, which depict mouse click hotspots, we can infer some information. The form elements are clicked the most as expected, the zoom buttons are popular, which makes sense, because they require multiple clicks. Finally, users seem to click on the time series metric dropdown a lot, which might be the problem we discussed in Table B.2. We can also see the fullscreen buttons being used a couple of times. The second search form is hardly used, which confirms our findings from previous results that the user can query a second word is not that obvious.

Click Counts Per Session For Items With More Than 1 Click and **Mouse Over Count Per Session** tables are omitted because they contain discrete count statistics, which are incomplete without the full sessions being captured.

In Figure B.5 we can see that there is mouse movement everywhere on the screen, where there is an element. It is again evident, that the users interacted with the first network graph. The second query part of the search form is hardly used and the movement on the ego network graph is spread out the whole component and not divided into two as top and bottom. This points to users only visualizing the first graph and interacting with

the result, because it covers the whole component. Only then, when the user visualizes the second graph, the component gets divided to top and bottom.

In Figure B.6 we can observe that users seem to drag the nodes everywhere. The second query not being used is evident here as well, because the drags are all in the middle and not divided into top and bottom.

The metrics in Table B.9 unfortunately are also not reliable because of the session disconnection bug. However it is interesting that scrolls have massively increased. This might be the direct result of the zoom buttons being more visible. There is also a new stat called resize count, to see if users use the resize functionality. Clicking on fullscreen buttons is not considered part of resize functionality.

5.2.6 Summary

In Tables 5.8 and 5.9, we can see what usability problems and bugs were discovered by manual usability testing and by SAUTO. Unfortunately, because of the session disconnection bug, SAUTO results are partially accurate and some results could not even be considered, because they contained discrete counts that required full sessions to be meaningful. Manual tests were able to discover 15 problems, while SAUTO was able to discover 5, with 3 of them not being discovered in the manual testing. SAUTO also discovered 5 cases of correct usability, which can be seen in table In Table 5.5. These numbers of SAUTO results are as explained before not the whole picture of the iterative phase, so they can't be taken as the fully accurate truth. But we tried to infer as much information as possible with the circumstances.

Table 5.8: Iterative Phase: Manual Usability Test Results

Description	**Category**
Right click is not utilized: visualize ego network by right clicking on a surrounding word node	Usability
Sometimes, it loads same data for different targetwords	Bug
Some users didn't like the colors, the though it was not clear what node colors indicate	Usability
Autocomplete cannot deal with special characters and doesn't print error message	Bug
Some users couldn't find +(second query) button	Usability
Year slider was seen quite late	Usability
Some words remained when user clicked on deselect-all on parallel coordinates	Bug
The tool was not responsive for smaller screens	Usability
Info page was too long	Usability
Table View selection was gone after revisiting info page	Bug
Sometimes the first ego network visualization pane is cut into half of the normal size even though second visualization is not active	Bug
Paralle lCoordinates, Timeseries, Table view update without clicking on 'Visualize'	Usability
Time series graph was showing only one line althouth two targetwords were selected	Bug
It was not intuitive enough for some users how to close the ego-network visulization	Usability
Some users didn't realize that there was the fullscreen buttons	Usability

Table 5.9: Iterative Phase: SAUTO Usability Test Results - Problems

Description	**Category**	**Problem found in**	**Found also in Manual**
Info page is too long	Usability	Metric based	Yes
Too many components in the results for beginners	Usability	Metric based	No
Second query and the network graph are used much less	Usability	Metric based, Pattern matching, Inferential	Yes
Too many clicks on info (either too much text or users trying to click non interactive images)	Usability	Pattern matching	No
Timeseries dropdown menu has too many clicks	Bug	Pattern matching	No

Table 5.10: Iterative Phase: SAUTO Usability Test Results - Correct Usages

Description	**Problem found in**
Users spend some time on the info page reading as expected	Metric based
Network graph is interacted with by most users	Pattern matching
Zoom buttons are used efficiently	Pattern matching, Inferential
Table view is used by most users and is intuitive	Pattern matching
General workflow order of the users match developer expectations	Pattern matching

5.3 Summative Phase

The Summative phase was the third and the last round of usability tests for the DYLEN tool. Unlike the first two phases, there were no manual usability tests. There were no controlled environment and no specific user tasks given to the user. There were in total 112 sessions collected in this phase, with all of them being real life users, who used to tool in their own time and in unspecified locations (location info was not captured). As explained in the chapter Approach, while formative and iterative phases are based on quality, summative phase is based on quantity. Therefore a much larger of sample size was gathered in the 112 sessions.

After the iterative phase, the developers worked on new features, usability improvements and bug fixes, just like after the formative phase. The results from the iterative phase of both manual usability tests and SAUTO were discussed inside the development team. After the discussion, the results were prioritised, with focus being on fixing bugs discovered during testing and fulfilling the last remaining feature requirements before launch. Usability problems were also considered when putting the finishing touches. The following are the main changes developed after the iterative phase tests:

- Changed the location of the search forms from the top to the left
- Added a feature to be able to right click nodes to view exact node metrics
- Right click on node context menu also offers a “Set as target word” button, to speed up querying for words in ego networks
- Changed reset query button from icon to text to make it more visible
- Each info page tab was separated into collapsables to add more structure and it was expanded with example use cases
- Implemented General networks: added two new pages: General Network (Party) and General Network (Speaker) in addition to already existing Ego Network
- Added spinner to Visualize button to indicate network loading
- Made settings start expanded when a network is visualized (as opposed to users having to open it manually)
- Added table view for time series component
- Implemented filtering on tables
- Fixed first two columns of the table in place
- Added option to enable/disable cluster colors

- Made second search form automatically show up when the first form is searched (instead of users having to press the plus button)
- Implemented timeout warning for queries that take too long in general networks
- Made the tool more responsive to cover lower screen sizes
- Improved accessibility of the tool
- Added theme to the tool: different colors than white and button icons (info and settings)
- Added info and settings buttons for each component
- General bug fixes

5.3.1 General Results

In Table 5.11 the updated user stories with their relevant element ids can be viewed. There are many new stories. *"Basic Word Search"* and *"Graph Comparison'* have been divided into three for each page(ego and general networks). *"Table View Visualization"* has been divided into two for node metrics table and time series table. *"Settings Button"* was also divided into two, because users can either click the general settings or component specific settings. In addition to those separations, there were also the following new user stories: *"Select word from node context menu"*, *"View node specific stats (open Context Menu of node)"*, *"Use filter on node metrics table"*, *"Use filter on time series table"* and *"Compare Time series"*.

Table 5.11: Summative Phase: User Stories to Element Ids

User Story	**Element Id**
Basic Ego Network Word Search (Visualize a word)	queryButton-Ego-pane1
Basic General Network Party Search (Visualize a party)	queryButton-Party-pane1
Basic General Network Speaker Search (Visualize a speaker)	queryButton-Speaker-pane1
Ego Network Word Graph Comparison	queryButton-Ego-pane2
General Network Party Graph Comparison	queryButton-Party-pane2
General Network Speaker Graph Comparison	queryButton-Speaker-pane2
Select word from node context menu	context-menu-select-as-targetword
View node specific stats (open Context Menu of node)	right-click-pane1-node
View node specific stats (open Context Menu of node)	right-click-pane2-node
View Node metrics Table Visualization	node-metrics-table-button
View Time series Table Visualization	time-series-table-button
Use filter on node metrics table	node-metrics-filter
Use filter on time series table	time-series-filter
Visualization (Select a word in the graph)	pane1-node
Visualization (Select a word in the graph)	pane2-node
Select Table Item	table-item
Visualize a different year	year-slider-pane1
Visualize a different year	year-slider-pane2
Sort Table	table-sort
Remove line from Parallel Coordinates	parallel-coordinates-x-button
Settings Button	settings-button-nav
Settings Button	settings-button-result
Compare Time Series	timeSeries-relative-option

The best and the worst performing elements do not require much interpretation. As explained in the section Description of SAUTO Results, they are the result of aggregating all the analysis methods. Analysing the rest of the results confirms the best and worst performing elements. Especially, all three of the best performing elements are the most used elements of the tool, which can be confirmed from the rest of the results. On the other side, there are interesting items in the worst list. It is clear why the number one worst element is “time-series-filter”, because of all sessions, not a single user used it. It is also a similar case with “right-click-pane2-node” and “selectSpeaker-pane1”. There were just too few sessions in which these elements were interacted with.

Best Performing Elements:

1. year-slider-pane1
2. info-tab
3. pane1-node

Worst Performing Elements:

1. time-series-filter
2. right-click-pane2-node
3. selectSpeaker-pane1

5.3.2 Metric Based

Table 5.12 shows some general metrics about all the sessions in the phase. There were 112 sessions in this phase: a much larger sample size than the previous phases. There are also more unique user stories, as many new ones were implemented and old ones were divided into multiple stories. User story count completed per session has decreased compared to the formative phase, along with average session duration. Additionally, average user story completion duration have increased. All this is due to the fact that in the formative phase, there were specific user tasks given to the testers, who had to complete all of them, which took around 30 minutes on average. However in the summative phase, there were no such user tasks given to the users and they visited the website on their on time, so there was no extra incentive holding them on the page as long as the users in the formative phase. This is to be expected, because we can't expect every single user to be fully dedicated to testing out all the features of our website. When analyzing the rest of the results below, we will keep this in mind.

Table 5.12: Summative Phase: General Metrics

Metric	**Result**
Eligible Session Count	112
Possible Unique User Story Count	19
User Story Completed Per Session	11.81
Average User Story Completion Duration (s)	40.89
Average Session Duration	12m59s

In Table C.1 it is clear at a first glance, that each user completed much fewer stories per session. To compare with the formative phase, *"Visualize a different year"* on the first pane was the top with 7.33 occurrences per session, while the top in this phase is the same story with only 2.11 occurrences per session. The reason for this decrease in numbers was explained above. The second most occurring story *"Visualization (Select a word in the graph)"* didn't change as well. The third, fifth and sixth stories in the table are the same story in different pages, namely *"Basic Search on Ego/General Network"*. The fourth element is of importance as it is *"Compare Time Series"*. According to the numbers, every session time series was compared once but rest of the features in the time series component weren't interacted with as much. This points to us that time series component is not well understood by users. The rest of the stories are all completed less

than once per session, which tells us they are not useful/intuitive to new users at a first glance. We dig deeper in the following paragraph.

Many users do not use the second search form to compare two different queries, as evident from the comparison stories having low occurrence counts (especially less than 1 per session). However, *"Visualize a different year"* on the second pane is completed a fair amount of times, although, graph comparison feature wasn't used that much. This confirms that the year slider itself is a success in terms of usability. *"View node specific stats (open Context Menu of node)"* seems to do badly as it is completed roughly once in every third session. But this can not be regarded as solely negative, because there is no visual cue to right click having a special context menu in the tool, so it is fine that only some users seem to figure that out intuitively. However, they seem not to use the *"Select word from node context menu"* functionality. As we come into less than 25% territory, we can all consider these stories as not very intuitive and in need of a rework. Table views of both node metrics and time series are used barely, which was not the case in the formative phase, because there were specific tasks requiring the users to interact with them. Summative phase shows us that the existence of table views is not obvious to the user. This is a good example of the use of SAUTO, because we can conduct usability tests in an unmoderated real life scenario. Finally, as a result of table views not being used, we can see that only a handful users completed the story *"Use filter on node metrics table"*, while none completed the story *"Use filter on time series table"*. After seeing these results, we realized that there was a bug in the time series filter, which prevented it from working at all. So this can be viewed as a bug, discovered by SAUTO.

In Table C.2 we can start by ignoring the first story, which was not once completed and therefore can not have a completion time, which is why it sits at 0. The next story is *"Select word from node context menu"*. The reason it is high up on the list is that this story can be completed right after the story *"View node specific stats (open Context Menu of node)"*. So its duration always starts right after when a user right clicks on a node. In that case, we can't really regard this as a positive or a negative, because it is the first thing the user sees, when they right click on a node. Next two stories concern the table. We know that not many users even viewed the table but it seems, the ones who do know its features well enough to use it fast. We can call these users power users. Same thing goes for most of the top stories in this table: *"Visualization (Select a word in the graph)"* in the second pane, *"View Time series Table Visualization"*, *"View node specific stats(open Context menu of node)"*, *"Visualize a different year"* in the second pane. Most users don't complete these stories, but the ones who do, know enough about them to use complete them fast. There is one story that comes high in the list just as in the previous table, which is *"Compare Time Series"*. It seems again, we can conclude the users like this feature, even though they do not try the other features of the time series visualization. Towards the end of the table, we see the *"Basic Network Search"* stories, more so in the first pane than the second. This works as expected just like in the formative phase, because completing a basic search story requires filling in the search form first, which takes some time. Last story in the table is *"Settings Button"* in the top

corner of the screen. It seems users generally go to the settings when they are stuck and don't know where to go. This is an expected behaviour and is something the developers want, because the default settings should be optimal and the users should change them only after they try out the defaults.

In Table C.3 the first story we see is the time series table filter one, which we know no one completed and can be ignored. The second story however is *"Select word from node context menu"*, which astonishingly requires zero interactions to complete. This is however the same phenomenon we discovered in the previous table. When the user right clicks on a node, the only button in the context menu is the one they click on directly. If they decide to click on somewhere else, the context menu disappears. Since the calculation starts from the last user story, which is triggered by the right click itself, it is always zero. It looks like the next elements in the table perform well: *"Use filter on node metrics table"*, *"Visualization (Select a word in the graph)"* and *"View node specific stats(open Context Menu of node)"*. However, we know from previous tables that not many users completed these stories. Only a handful of power users did and did so efficiently. Therefore we can not view these stories positively from this table even though the numbers look good. Noteworthy on this table is the fact that six of the last seven stories are *"Basic Network Search"* stories. It is again the same fact that users need to fill out the search forms, which involves many interactions on the users part.

With a first glance on Table C.5, we can see the amount of elements has massively increased between the phases. This number correlates greatly with the amount of new features the developers built in between the last two phases.

We can observe the first three items are again of info page. Just as the previous two phases, this is the first screen that the user sees and logically the first components they discover. Next one is the ego-network-tab, which is the first of the main three tabs on the top and the best way for the users to start using the tool. The next item is the left-query-bar, which is where the search forms are contained, which is also the next logical step for the user. The next couple of items are again the result of the same power user phenomenon, which means power users find elements faster but because they are a very little percentage of the users, we can't regard these items as good performing. The rest of the table paints a clear picture of the general workflow of the user. Firstly, they visualize a word in the ego network tab, then they move on to general network party tab and search a party there. They play around with the graph some more by moving the nodes around and trying the zoom buttons. They discover the year slider around this time. Afterwards, their attention slides to node metrics and time series components including their table views on the right of the screen but it doesn't stay there long. The users try out the general netwok party speaker tab next. They fill in the search form and and visualize a speaker's general network. They play around with the graph, then the node metrics and time series visualizations and close the session around that time. Settings menu is opened and some settings changed throughout the session without a particular order. In general, as to be expected, the second pane in each network page is discovered later and interacted with much less than the first graph. In this sense, we can

see that there is a still room for improvement for making the comparison more intuitive to the users.

5.3.3 Pattern Matching

In Table A.5 we need to point out that the minimum repetition count in this phase has been set to 50 in the table in this work. The full table with the minimum being 5 can be found online at the SAUTO analysis page. 5 as a minimum number made sense for the first two phases where the sample size was very small compared to the summative phase. Since we have more users and therefore more sessions in this phase, a different minimum was needed to avoid having repetitions of negligible occurrence counts in relation to all the 112 sessions.

When we look at the table, we start by looking at the top three items being in the info page again. This was a phenomenon that happened in all the previous phases. There were two reasons for this in the previous phases: images looked interactive and users tried to click on them and the text was too long, so users had to highlight the text. By putting images in clearer image boxes with navigation arrows to change screenshots, the developers tried to fix the first issue. By dividing the info page into collapsables and tabs, the developers tried to alleviate if not fix the second issue. The text was even extended to include more information but the tabs and collapsables provide some hierarchy. The fact that info-tab has more repetitions than info-tab-content (text) and info-collapsable-button having a similar number of repetitions to info-tab-content shows that the efforts of the developers were fairly successful. Next item is the zoom in button which is the same as previous phases, and it is expected. year-slider-pane1 is the next most repeatedly clicked item which tells us, that the users change years to quickly view and compare different years. Most of them are not aware that the second pane is there not only to compare words but also to compare years within the same word. pane1-node is another reminder that the users play around with the nodes, which is what the developers want. The next item in the table is a somewhat confusing one: timeSeries-relative-option. There is no apparent reason for this item to be repeatedly clicked, so this should be inspected by the developers as a potential usability problem or bug. The same can be said about the timeSeries-metric-option, which also has many repetitions.

In Table C.7 the cutoff minimum is set again at 50 for the same reason as the previous table. Again, the full table can be found online at the SAUTO analysis page. All the items on this table point to correct usages of the tool in expected order. There are two sequences in this table: First one involves the user filling in the search form and visualizing a word in expected order. The second one is when the user clicks in on the tabs and the collapsables in the info page.

In Table C.8 the cutoff minimum is set at 30 for the same reason as the previous tables. This table gives the same exact information as the previous table. The first and the last elements are visualizing a word in the first and the second panes. From the count numbers, we can see that the users visualized the first word almost four times more than

the second pane. The four items in between are the sequences of clicking around in the info page.

In Table C.9 the cutoff minimum is set at 20 for the same reason as the previous tables. Same as the previous tables, this one also has repetitions involving the same two scenarios. The first three items are the info page clicks and the last three items are the different variations of filling out the first search form and visualizing a word in the ego network.

In Table C.10 the cutoff minimum is set at 15 for the same reason as the previous tables. The first two elements show an interaction order of visualizing a word, changing the year and then clicking on a node. The third is our coveted graph comparison story. We can observe that some users were able to utilize the comparison feature, but not as much as the developers would like. The rest of the patterns show the same scenario of searching for a word or party or speaker and then changing the year or playing around with the resulting graph by moving the nodes around.

In Table C.11 the cutoff minimum is set at 10 for the same reason as the previous tables. Here, we can see the users visualizing a word in the ego network and either directly changing years or directly playing around with the nodes. The last item on the table is the combination of the first two, which is changing the years and then playing with the nodes.

In Table C.12 the cutoff minimum is set at 10 for the same reason as the previous tables. The only item confirms what the previous tables confirmed. The users visualize a word and then keep changing years, instead of visualizing the same word in the second pane and comparing different years side by side.

5.3.4 Task Based

In Table C.13 we can observe the specific expectations of the developers compared to what real users actually did. We go over each expectation specifically below.

Analysis of Developer Expectations

1. 45 is a significant amount of people who haven't tried out zooming in/out. Developers should look into making the zoom buttons even more visible.
2. The expected percentage target wasn't reached. This is definitely a power user functionality, which should be made more visible
3. Same as the previous, developers should make the DYSEN link (link to the sister project on the top header) more visible.
4. This one seems to be a huge failure and general network search form needs to be redesigned.
5. This one is a failure as well. Not one single user downloaded the raw data of the tables, but this can be attributed to the fact that tables weren't used much at all.

6. This one is correlated with no 4. Users generally didn't interact much with the general network search form, which poins to a usability problem.

7. The fact that settings options were not changed much might mean that the default settings were fine or the users didn't bother with settings at all. The second one seems to be the case with our knowledge of the rest of the results (mostly beginners instead of power users skimming through the tool).

8. This expectation is generally fulfilled and confirmed from other results C.5, A.5 as well.

9. Fullscreen buttons were not visible enough for most users.

10. We established that info page was fairly a success and we can see that most users interacted with it as expected.

11. Reset button is not a main attraction and the results show it. Most users weren't interested in searching new queries it seems.

12. This one is split into half, which is not that expected. It might be that half of the users ignore the dropdown menus accepting the defaults and focus directly on the target word, which is expected behaviour but not optimal.

13. This one is another failure, select/deselect all interaction has been improved between phases but users still seem to struggle with it.

14. Just like expectation no 7, this is the result of most users not bothering with settings, instead of defaults being good.

15. Another item that should be made more visible or intuitive to the user.

16. Even though users seem not to notice the fullscreen buttons, they found the resize functionality. This can be regarded as a successful feature usability-wise.

17. Time series component was generally not used much but more than half of the users seem to at least take a glance at it. The whole component should still be made more intuitive.

18. These metrics are targeted to power users and the numbers show that most users ignore the this metric option

19. Since most users didn't bother with changing general network default queries, they didn't get any timeouts, which only occur if the percentages of nodes are increased.

20. Parallel coordinates were also not paid attention to by most users and it shows in this metric as well.

21. It is a failure that only half the users even hovered over the parallel coordinates. Because it is a unique visualization, it is understandable that most users didn't understand it but that is why it is important to make it even more intuitive than it is.

22. Only 5 users tried out this functionality. All the previous results like completion duration and completion interaction stats showing this feature in a good light can be ignored, because of the small sample size.

23. Same conclusion as before: table not viewed much, filter used even less.

24. Less than one fourth of the users seem to have found the table view. This is a big usability problem, because it contains a lot of useful functionality.

25. Less than half the users played around with the graph. This tells us that some user do not realize that the graphs are interactive.

26. Even fewer users played with the second pane. This is a huge failure.

27. This element is another failure. Most users didn't understand parallel coordinates and therefore most didn't use it.

28. Half the users didn't try out ego-networks, assuming they tried out general networks, they are still missing out on core functionality.

29. Half of the users that did try out ego networks tried to compare two. This is a failure and needs to be made more obvious and intuitive.

30. Similar to ego-network, half of the users tried general network party. This is fine but still means they are missing out on core functionality.

31. Same as ego network, second pane is used much less.

32. Same story as ego network and general network party, users should theoretically try all three tabs. So this can be regarded as a usability problem.

33. Second pane is again not used much.

34. Right click functionality is not obvious at all for users and needs to be made more explicit.

35. This tells us the same thing as the previous item on this list.

36. Our theory that most users didn't bother with settings is confirmed here. This feature is targeted more towards power users, so it is fine.

37. This tells us the same thing as the previous item on this list.

38. Table views were used very little and this is a big usability problem.

39. This tells us the same thing as the previous item on this list.
40. This feature is buggy and doesn't work, so 0 is expected.
41. Table view is again lost on most users.
42. This story came up in the mouse click repetitions table A.5. We see that not many users tried it. The fact that it is high on the mouse click repetitions table point to a usability problem with this element.
43. Half the users didn't change the year. Other results show that the year slider is a very well performing element usability-wise. The fact that only half the users tried it out can be attributed to users not being interested in it rather than it being not intuitive.
44. The same conclusion can be made here as the previous item on the list.

Overall, according to the results, the developer expectations seem not to be met. However we need to rethink our analysis of the developer expectations. The users of this phase were real life users, so they did not use the tool after being given instructions like in manual usability testing. They did not have the same motivations as the users who participated in usability tests in the first two phases. That is why, we can not expect all of them to use all the features of the tool. So the developers should definitely take that into account when analysing the results of the developer expectations table.

5.3.5 Inferential

From Figures C.1 C.2 C.3 C.4 C.5 C.6, which depict mouse click hotpots, we can confirm some of our conclusions from the previous results. The most clicked part of the screen is the search forms on the left, with the first form being clicked on more. The top tabs also have some difference between them. The General Network (Speaker) is clicked on less than the other two. The info button on the top right is a big hit, as users switch between results and info page. Repeated clicks on the zoom buttons on the first pane (when it covers the whole component vertically) are also visible. Node Metric Comparison and Time Series Analysis has much fewer clicks. The top left of the screen is also very red, because the logo there serves the same purpose as the info button. This feature was added after the iterative phase.

In Table C.14 we can see again that the top three most clicked elements are the info page. Next come elements in the search form for the first pane in the ego network. The top tabs are clicked more than once per session as without them, the users wouldn't leave the info page and wouldn't use the tool at all. It is again worth noting that the numbers have fallen compared to the formative phase, because the sessions during those tests took longer, because testers had specific instructions.

In Figure C.7 we can see that basically everywhere is covered by the users. There are a small amount of patches without any mouse movement but there are no elements there either, so everything is fine usability-wise, from what we can infer from this hotspot map.

In Table C.15 same as the formative phase, the users go mostly between the search form on the left and the results on the right. Info tab comes next and then the tabs on the top headers. So overall, there is nothing out of the ordinary in this table.

In Figure C.8 the first thing we see is a giant blob of arrows in the network graph component. The fact that it is not spread out more but is confined mostly to that component tells us that the fullscreen functionality was not discovered by most users. The couple of arrows leading to the top left stem probably from users dragging nodes out of the screen. There is one more little detail on this hotspot map, which is that not one single user discovered the drag functionality under the settings. Under parallel coordinates options, the user can determine which metrics should be displayed and can arrange their order by dragging them up and down. This functionality is targeted at the most hardcore power users and it seems there was no such user.

In Table C.16 we can observe that the users spend most of their time on the info page. This is a positive and a negative. The positive is that they take their time to read the information, which helps them fully utilize the features of the tool but it is negative because the information is too long and it takes time away from the real usage of the app. The order of ego-network-tab, general-network-tab (party) and lastly general-network-speaker-tab is a confirmation of previous results. It also corresponds roughly to the numbers in the developer expectations table C.13, namely the expectations from 28 to 33.

Table C.17 gives us an important hint that the most interaction happens on the ego network page. This page has the most features and is the first tab. Also general network (party) and general network (speaker) are very similar, so they steal attention from each other. So this relationship between the pages is to be expected.

From the metrics in Table C.18, we can see that the scroll count has decreased massively compared to the previous phases. This has two causes: users spending comparatively less time on the tool, and the zoom buttons are used more. Resize count has increased compared to the iterative phase, which is a good sign. Key press count has also decreased substantially. This has again two causes: users spending comparatively less time on the tool, and the auto-complete feature working better. Finally, timeouts were a new feature that was introduced in this phase as too large general networks could cause crashes. However, we see that not many users encountered these timeouts because as we know from the previous results, not many users changed the search form in general networks at all.

5.3.6 Summary

In Table 5.13 we can see what usability problems, bugs and correct usages that were discovered by SAUTO. There were in total 18 problems and 14 examples of correct usages

discovered by SAUTO.

There were two additional observations, that were not part of official SAUTO results. We live in an age where most online tools offer mobile versions and some users expected the same from the DYLEN tool, which was not the case. The tool was not optimized for mobile phones and did not show the results at all. Additionally, touch screen interaction is different than a keyboard mouse interaction that SAUTO is based on. From server logs, we were able to determine that there were at least 53 sessions, where users tried to view the tool on their phones (which was not included in SAUTO results). Secondly, there were 245 sessions, in which the users stayed less than 3 minutes on the tool and therefore were not considered for the results by SAUTO. This is double the number of considered sessions and tells us that SAUTO loses around one third of its users too quickly. Finally, we can see from the results that there were a handful of power users, who used most of the functionality of the tool, while most users were beginners and tried out only a part of the full functionality.

When studying the results of the summative phase, we should keep in mind the differences between the first two phases and the summative phase. As explained before, the first two phases were also manual usability tests. The tests were unmoderated (unless users got absolutely stuck) and in a controlled environment (at home in an undisturbed room in an online call). The participants were given clear instructions and were observed while using the tool. There were in total 11 participants in the first two phases. In the summative phase however, there were no instructions and no controlled environment. The users used the tool in their own time, in unspecified locations. There were in total 112 sessions with a possibility of some users trying the tool multiple times. Manual tests with 100 participants is unthinkable and that is where SAUTO shows its true power. The difference between the sample sizes directly affects the meaningfulness of the results.

Table 5.13: Summative Phase: SAUTO Usability Test Results - Problems

Description	**Category**	**Problem found in**
Time series is not used by most users	Usability	Metric based, Task based, Inferential
Second search form and its results are used much less compared to first form in each page (graph comparison not used by most users)	Usability	Metric based, Pattern matching, Task based, Inferential
Table views are not used by most users	Usability	Metric based, Task based, Inferential
Time series filter does not work properly	Bug	Metric based, Task based
Parallel Coordinates is not used by most users	Usability	Metric based, Task based, Inferential
Time series dropdown options have too many repetitions	Usability	Pattern matching, Task based
Zoom buttons were not discovered by enough users	Usability	Task based
DYSEN link is not obvious	Usability	Task based
General Network Search Form is not used by most users	Usability	Task based

Table 5.13: Summative Phase: SAUTO Usability Test Results - Problems - continued

Description	**Category**	**Problem found in**
Fullscreen buttons were not used by most users	Usability	Task based, Inferential
Reset query button was used much less than what developers expected	Usability	Task based
Select/deselect buttons are not used by most users	Usability	Task based
Show clusters button is not used by most users	Usability	Task based
Right click to show context menu was not used by most users	Usability	Task based
Some users do not realize that network graphs are interactive	Usability	Task based
Most users try only one type of network instead of all three	Usability	Task based, Inferential
Parallel Coordinates node metric order wasn't changed	Usability	Inferential
Info page takes too much time of the user's attention	Usability	Inferential

Table 5.14: Summative Phase: SAUTO Usability Test Results - Correct Usages

Description	**Problem found in**
Year slider is used substantially by most users	Metric based, Pattern matching, Inferential
Users look at settings only when they don't have anything left to do and don't interact with it much as expected	Metric based, Task based
A small amount of power users seem to complete tasks efficiently	Metric based
General workflow order of the users match developer expectations	Metric based, Pattern matching, Inferential
Info page tabs and collapsables are interacted with by most users as expected	Pattern matching, Task based, Inferential
Zoom buttons are used efficiently when discovered	Pattern matching, Inferential
Interactivity of network graph is discovered by half the users and when discovered it is used extensively	Metric based, Pattern matching, Inferential
First user story completed is visualization as expected by the developers	Task based
Resize functionality is used by most users	Task based, Inferential
Most users didn't get a timeout in general networks	Task based, Inferential
Info button on the top right works well	Inferential
DYLEN logo as info button works well	Inferential
Most interaction happens in the ego network page, because it has the most features	Inferential
Auto-complete feature was improved	Inferential

5.4 Interviews

As a qualitative research method, interviews were conducted about SAUTO and semi-automated usability testing with the two developers of DYLEN (denoted as Developer 1 and Developer 2), who had first hand experience with SAUTO and an external usability expert (denoted as External Expert). The interviews were standardized open-ended interviews as defined by [26], in which the questions were identical for all participants and were all open ended. Some questions were not applicable to the external expert, since he did not take part in the DYLEN project. His answers to those questions are marked as *Not Applicable*. All three interviews were conducted through online forms to give the interviewees enough time to think and formulate their answers in their own time. The answers were copied exactly as they were written below (except the hint about Can, the author of this thesis). The interpretation and the evaluation of the interviews can be found in the chapter Conclusion.

1. *Age:*

   Developer 1: 18-30

   Developer 2: 31-40

   External Expert: 18-30

2. *Gender:*

   Developer 1: Female

   Developer 2: Male

   External Expert: Male

3. *Education:*

   Developer 1: Bachelor's Degree

   Developer 2: Master's Degree

   External Expert: Master's Degree

4. *What experiences do you have with human computer interaction, human centered design, usability testing and/or UI/UX design?*

   Developer 1: Worked as a tutor for an HCI course at university, used human centered design & usability testing in my own projects.

   Developer 2: Some experiences

   External Expert: I work as a cross-over between a Software Developer and UI/UX-Designer. As part of the computer science teaching team at the University of Vienna, I teach human-computer-interactions, of which usability is a centrepiece.

5. *How long were you part of the DYLEN team (and worked with SAUTO during development)?*

   Developer 1: Since May 2021 (9 months)

   Developer 2: Since Feb 2020 (24 months)

   External Expert: *Not Applicable*

6. *Have you ever conducted or taken part as a participant in manual usability testing (not considering DYLEN)? If yes, what was your role and how was your general experience?*

   Developer 1: Experience as a participant: excited and nervous in the beginning, afraid of doing something wrong, tried my best to make useful suggestions and give helpful feedback. Experience as an interviewer: good preparation is the key, once you have your participants and can start with the actual test, it was fun and gave very useful insights.

   Developer 2: Yes, silent observer

   External Expert: I have conducted manual usability studies as well as participated in studies.

7. *Do you have any experience with semi-automated usability testing (not considering DYLEN)? If yes, please elaborate.*

   Developer 1: No

   Developer 2: Yes. I had another project.

   External Expert: Nope

8. *During development of DYLEN, did you find SAUTO easy or hard to work with? Please elaborate on your answer.*

   Developer 1: Easy, Can (the author of this thesis) did most of the SAUTO work ;) Trying not to get into conflicts with existing SAUTO attributes or methods was fine.

   Developer 2: It was hard, because SAUTO was actively in development. I can also imagine that using SAUTO for other projects could be hard because of additional workloads.

   External Expert: *Not Applicable*

9. *Did you find SAUTO usability results (problems and correct usages) useful? Why or why not?*

Developer 1: I found them useful, but sometimes hard to interpret. More visualizations would have been nice. Without Can's (the author of this thesis) explanations of some metrics, I might have misinterpreted some of the values.

Developer 2: Some of the insights were useful.

External Expert: The results can certainly help improve the usability of software projects. The repeated patterns are a great way to track how users move through their task. The repeated user stories offer a more high-level view of how user stories relate to each other. The metrics offer a great overview of how your product performs when it comes to usability. The correct usages also offers the product team a chance to see how well they researched their users.

10. *Would you replace manual usability testing with SAUTO (or semi-automated usability testing)? Why or why not?*

Developer 1: Depends on the application to be tested, but for most applications, I would stick to manual usability testing (or ideally use both) because the qualitative feedback is more important to me.

Developer 2: No, manual usability tests is able to capture issues that can be missed by SAUTO, and also seeing testers live in action is valuable.

External Expert: I think that it can certainly offer great insight into the usability and is definitely an option for teams that have little to no resources for doing manual usability testing. Perhaps I would use it to gain an overview of how my product is performing and using the data I would identify features and elements that need improvement.

11. *Would you recommend SAUTO (or semi-automated usability testing) in your future projects? Why or why not?*

Developer 1: Yes, because it has a high potential of providing useful insights. But I would suggest to use it in addition to manual usability testing.

Developer 2: If SAUTO gets more easier to use with minimum workload, then yes.

External Expert: I would definitely recommend SAUTO and given the option, would love to implement it in one of my future projects. The data and insights gained are very valuable and have require less resources than manual usability testing.

12. *Were there any disadvantages/negatives of using SAUTO during development? If yes, what?*

    Developer 1: (Small) overhead of making sure that SAUTO ids and methods were used correctly and the fact that single components were hard to integrate (year slider for example).

    Developer 2: There were minor issues constantly, until 2nd round of usability tests, but it's because it was in development. Dealing with sauto-id could be cumbersome.

    External Expert: *Not Applicable*

13. *What kind of usability results/approaches/metrics (that weren't provided) would you suggest for SAUTO, if any?*

    Developer 1: A timeline visualizing the order of tasks/actions; more visualizations instead of just tables.

    Developer 2: I think applying machine learning based on user actions to generate user personas/clusters might be interesting.

    External Expert: Interesting would be to provide an option to input tasks that users have to perform, similarly to the way you would go about manual user testing. So when setting everything up, a user would receive a prompt with a task that they have to complete. When it comes to the heat maps and screenshots, maybe have each screenshot display with its corresponding heat map. That way the results are a little easier to read. An interesting approach would also be to log what users type in as a text when searching.

14. *Which SAUTO results did you find most useful, why?*

    Developer 1: Pattern matching, repeated patterns, the heat map for clicks

    Developer 2: The heatmap and also the statistics compared to the developer expectations were useful.

    External Expert: Most useful for me form a product perspective probably the mapping of the developer expectations to the actual occurrences. Alongside that definitely the repeated patterns and repeated user story patterns are very valuable when it comes to testing this way, because you do not actually see your users clicking on buttons. The user story duration is always a great indicator of where users are struggling to complete their tasks.

15. *Which SAUTO results did you find least useful, why?*

    Developer 1: The mouse movement hotspot map, because (almost) everything is yellow and it shows the results for all sessions overlapping.

An option to select single sessions and/or have a threshold for frequently visited areas would be nice.

Developer 2: I think most of the statistics and results have some kind of value.

External Expert: The mouse drag map was a bit crowded for the 3rd round of testing. The drag map was more clear when less results are displayed - maybe we could limit the amount of results shown - question is how do we decide which results to display. The mouse movement map is another visualization that provides less value when compared to the other metrics displayed.

16. *Were there any parts of SAUTO results you did not understand?*

Developer 1: I don't think so, at least not after Can's (the author of this thesis) explanations

Developer 2: No, we got explained in detail.

External Expert: Nope.

17. *Do you have any suggestions for improvement for SAUTO?*

Developer 1: More visualizations, a filtering option for sessions (e.g. by duration or number of interactions), potential future work: an interface for configuring developer expectations.

Developer 2: Make it easier to use, for example with a frontend library (generating unique id's automatically), apply machine learning, make results page more beautiful :D

External Expert: One option like I said would be to provide an option for users to prompt users to complete certain tasks. The visualisation of the data could be improved upon - perhaps with different visualisation and also an option to compare different sessions to see how different users interacted with the product on the same task. Like you (the author of this thesis) mentioned during the presentation, dividing your testers into different groups depending on their experience with the product can help identify different needs for different users. An interesting idea would also be to somehow capture how the wording of button labels and explanations affect the usability. One thing that can also be incredibly useful is A/B-Testing and with SAUTO this can be done at quite a nice scale.

18. *Do you have any other thoughts you would like to share about SAUTO?*

Developer 1: Very nice tool! I think there is high potential for future applications!

Developer 2: Nice work :)

External Expert: The tool itself is really fascinating and is definitely a great way to gather data on the usability issues your product has.

CHAPTER 6

Conclusion

In today's software industry, it is essential for any software to go through usability testing. The software needs to be tested by real users so that the software's user friendliness and intuitiveness can be evaluated. However, conventional usability testing is a costly and time consuming process. To mitigate this problem, there has been research done to automate usability testing, in which the observers could be eliminated and the usages of real users could be captured and evaluated automatically, which we call semi-automated usability testing. In this work, we aimed to build up on that research by combining different approaches into one semi-automated usability framework.

We conducted manual usability testing and semi-automated testing during the development process of the DYLEN tool. The usability testing process was divided into three phases: formative, iterative and summative. In the first two phases, we conducted manual usability testing and semi-automated usability testing on the same users in the same sessions and compared the results. In the last phase, we conducted only semi-automated usability testing with more than one hundred sessions and compared the results to previous phases as well as qualitatively evaluated them through interviews.

6.1 Evaluation

The results suggest that manual usability testing can provide more usability insights than semi-automated usability testing when testing with a small number of testers. However semi-automated usability testing promises more usability insights with a large number of testers. Conducting usability tests with such a large number of testers is not possible. Additionally, semi-automated usability testing can provide a list of correct usages of the tested software, to confirm developer expectations.

In the formative phase, through manual usability testing with five users, we were able to discover 19 usability problems compared to 10 usability problems with semi-automated

usability testing. Out of these 10 problems discovered with semi-automated usability testing, only 3 of them were not already discovered in the manual tests. There were additionally 9 cases of correct usages discovered by semi-automated usability tests. These results suggest that with a low number of users, semi-automated usability testing can not replace manual usability testing but it can be complimentary to it.

In the iterative phase, we repeated the process with six users. Unfortunately, in the middle of each session, the SAUTO disconnected due to a bug and the rest of the sessions could not be recorded, This resulted in not getting full results from semi-automated usability testing. Even though we were able to discover some problems and correct usages from the semi-automated usability testing results, we could not infer any conclusive comparison between the two usability testing methods.

Finally, in the summative phase, 112 real life users tested the DYLEN tool in their own time and in uncontrolled environments. We were able to discover 18 usability problems and 14 correct usages from the results. This means there were a higher number of usability problems and correct usages discovered in the summative phase compared to the formative phase. Additionally, we found 5 usability problems, which were missed in the formative and iterative phases by both usability testing methods. There may be two reasons for this. Firstly, in the first two phases, users had clear instructions which encouraged them to use most of the features of the tool and when they got stuck, moderators helped. Secondly, the significantly higher number of testers in the summative phase can give more weight and confidence to the results. The results from all three phases indicate that while semi-automated usability testing can't fully replace manual usability testing, it can produce meaningful results when done side by side with manual usability testing and it can even produce better results after launch or the product to test with a high amount of real users, which is not possible with manual usability testing.

In the studies we went over in the chapter Approach, the authors evaluated their proposed solutions by comparing the numbers of problems discovered in manual tests versus their semi-automated test tools. Grigera et al. found half the results through their semi-automated usability tests compared to manual usability tests [20]. Their proposed tool additionally found problems that the manual test did not. These results match our results. The other studies' proposed tools' results were on par with their manual tests, as in they found a similar number of usability problems compared to manual tests. These do not match our results, which can be explained by various possible reasons. Kluth et al. limited their tool only to four predefined HCI design patterns, which correspond to four usability problems [7]. In this work, we did not limit our tool to any kind of predefined problems and aimed to find all possible usability problems. Baker et al. did manual usability tests and semi-automated usability tests with different users on the same software [5]. Bures et al. did two rounds of usability tests like our formative and iterative phases [9]. They however did not change the tested software, but adjusted their proposed semi-automated usability tool in between these rounds. They also did the rounds with 25 and 24 users each compared to our 5 and 6. These might be the explanation of the difference in results between our work and the listed studies.

To our best knowledge, no study evaluated their tool with many users without comparing to manual usability tests, which is what we did in the summative phase of this study. Since there were no examples of such evaluation, we used novel methods for evaluation in this context. We firstly compared the semi-automated usability results from the summative phase to the results from the previous phases. We then conducted qualitative interviews in order to evaluate the feasibility of SAUTO. The interviews were conducted with the two developers of the DYLEN tool, who worked together with SAUTO and an external expert on usability to get an unbiased opinion.

The interviews indicate that SAUTO was successful in its goal of applying semi-automated usability testing and providing meaningful usability results. The answers also indicate that there are a couple of ways it can be improved. Firstly, both developers were experienced in usability testing, which makes their answers valuable. However, only *Developer 2* had experience with semi-automated usability testing, which makes his answers extra valuable. *External expert* was well versed in usability as expected by their title in this work, which also give credibility to their answers.

Firstly, we should go over what problems both developers had with SAUTO and what it should/could have done better. From the answers to the 8^{th} and 12^{th} questions, we can conclude that SAUTO needs to be more developer friendly and should not get into the way of the development of the tested software. Another improvement for SAUTO would be to present its results in a better way, which is indicated by *Developer 1* in the answer to the question 9 and by *External Expert* in the answer to the question 13. This also relates to all three answers to the question 16. The presentation of the results was only clear after exhaustive explanations, so the presentation could be improved. Additionally, mouse movement hotspot map, and mouse drag hotspot map were found not useful by *Developer 1* and *External Expert* as indicated by their answers to the question 15. And in the question 14, mouse click hotspot maps, pattern matching tables and developer expectations were found to be the most useful results by all three interviewees. In the questions 13 and 17 *Developer 1* suggests more visualizations and more options for developers in SAUTO, *Developer 2* suggests using machine learning when analysing captured data, and *External Expert* suggests better visualizations, and session filtering options. Finally, in the question 10, all three interviewees agree on semi-automated usability testing not being eligible to be a full replacement of manual usability testing. *Developer 1* and *External Expert* additionally agree on semi-automated usability testing being incredibly useful and that they would suggest doing it together with manual usability testing. This conclusion confirms our conclusions from the quantitative results.

6.1.1 Research Questions

In this thesis, we answered the following questions:

a. Which resource and time consuming parts of manual usability testing can be eliminated via automation?

With our proposed framework, we achieved to eliminate the observer from the manual usability testing. The observer is a human resource and also their time is valuable. In eliminating the observer however, we added time and complexity for implementation of the tool.

b. How effective is semi-automated usability testing compared to manual usability testing?

Semi-automated usability testing with a large number of users has the potential to discover the same amount of problems and even problems that were not discovered during manual usability testing. Additionally, semi-automated usability testing provides information about correct usages, which manual usability testing doesn't focus on. However, manual usability testing has some important advantages compared to semi-automated testing. Manual usability testing with observers can go into more detail with only a handful of testers. Additionally, feedback from the testers can supply more insights to discover potential usability problems. Therefore we can conclude that semi-automated usability testing is not as effective as manual usability testing but it has its own advantages.

6.2 Summary & Contribution

In this thesis, we aimed to build up on the research about semi-automated usability testing by combining four general approaches into one framework. We conducted a case study, in which we developed a semi-automated usability testing tool (SAUTO) within the project DYLEN. We conducted two phases of manual usability testing together with semi-automated usability testing and one final round of only semi-automated usability testing. We also did qualitative interviews with the two developers of the DYLEN tool and an external usability expert. The quantitative results from all three phases and the qualitative interviews both suggest that while semi-automated usability testing cannot fully replace manual usability testing, it can be carried out as complementary to it. Additionally, it can be carried out with a much higher number of users than manual usability tests and it can also be carried out after product launch. These two advantages of semi-automated usability over manual usability testing makes it a viable option for development teams who want to carry out usability testing.

In this work, we proposed a novel framework for semi-automated usability testing, which combined four general approaches in one. To our best knowledge, there has been no tool or solution that combined all four approaches into one, they mostly used one or two. Additionally all the research into the topic of semi-automated usability testing we could find, evaluated their proposed solutions by comparing their results with manual usability testing with a small number of testers. In this work, we also carried out this method but additionally, we did only semi-automated usability testing with a large number of testers in the summative phase, which is not possible with manual usability testing. Finally, in addition to the quantitative evaluation of our proposed framework,

we conducted qualitative interviews, which also wasn't done in the scope of the research into semi-automated usability testing.

6.3 Limitations

Firstly, as discussed before, the iterative phase of this work suffered a bug and therefore could not be used fully. Having full results in the iterative phase would have been helpful to compare with the results from the formative phase and would have supplied us more information for side by side manual and semi-automated testing.

Secondly, the DYLEN tool is a software that has many specific and complex features and is targeted at a specific group, namely researchers in the fields of linguistics, political science and journalism. Finding testers for the target group in the summative phase was not easy. We can see from the results that most testers skimmed through the features and did not utilize the tool to its full extent. They were not motivated enough to fully explore the tool. If the tested tool was a more average software, for example, a shopping website, it would have been easier to find testers, who were interested in using the software to its full extent.

Finally, the development of SAUTO was heavily influenced by the DYLEN tool and its development process. SAUTO could only be implemented after the relevant features in the DYLEN tool were ready, which was the reason for a tight time schedule for the development of SAUTO, to have it ready for the tests on time. Additionally, because SAUTO was developed dependent on the DYLEN tool, it specifically caters to it. It would be a significant amount of work to adapt it to another tool and even more work to make it a universal plug-and-play tool.

6.4 Future Work

SAUTO as a semi-automated usability tool shows promise to be used complementary to manual usability tests. Additionally, it shows its full potential after the launch of the tested software, because data from a large number of users can be used to conduct usability tests. The development of SAUTO ran parallel to the development of the DYLEN tool. The frontend interaction capture code of SAUTO had to be built directly into the code base of the DYLEN tool. To be adopted in the software industry, SAUTO needs to be more practical. Specifically, it should not take more time from the developers to set it up. To solve this problem, SAUTO could be implemented as a stand alone tool agnostic library for the tool's preferred framework, which was VueJS in this case. Developers would just add attributes to their Html elements to register them for SAUTO to capture and everything could be done automatically. This could be a more optimal way to use SAUTO from the tested tool's developers' perspective.

To learn from our mistakes and to avoid repeating them, future work should add automatic and manual testing capabilities to SAUTO and the tested tool. Unit tests, integration

tests and end-to-end tests should all be implemented in SAUTO. Also, it should allow running mock usability tests so that developers can manually act as a tester and try out their tool, while debugging and making sure that SAUTO works as intended, as in connections are kept up, data is correctly transmitted, database is readable and writable etc.

SAUTO could also be improved in its analysis result presentation. Currently, it is a mostly non interactive bare minimum Html page. Results could be presented in a more smart way than just raw tables and images. An overlay of the tested tool can point out on a live tool, which elements cause problems. Some kind of debug mode of the tested tool, using SAUTO as a library could present information on all elements on the screen when user hovers over them.

To improve on the effectiveness of semi-automated usability testing, a variety of methods and approaches could be implemented in SAUTO. Some of these methods were already used in the studies described in the section Enhancing Usability Testing. There could be AI based machine learning algorithms analysing captured usage data. Another feature which could be combined into SAUTO is eye tracking and facial expression tracking. This data could provide helpful insights when combined with already implemented approaches in SAUTO. Webcams and mobile phone cameras could be used to that end. However, this would require permission from the user to make sure their privacy is respected. Users might not be willing to share their camera during their daily lives, when using the tested software. Automatic analysis techniques like facial emotion recognition could be applied onto captured video recordings. SAUTO analysis could also be combined with post-test surveys/questionnaires with users. These would help with quantifying user experience.

Another problem that could be solved in the future is the distinction of user types. In the results, we have seen that some users are power users and are able to complete user stories efficiently, while some users have a hard time grasping the tool's features. SAUTO could automatically distinguish between such users and have an option to display their results separately.

Finally, one more way to improve SAUTO analysis is having the automatic critic function. SAUTO could not only point out the difficulties but also propose improvements. That would be a huge step in the automation of usability testing.

APPENDIX A

Appendix: Formative Phase

A.1 Metric Based

Table A.1: Formative Phase: User Story Completion Count Per Session

Element Id	**User Story**	**Occurrence Count Per Session**
year-slider-pane1	Visualize a different year	7.33
pane1-node	Visualization (Select a word in the graph)	6.67
table-sort	Sort Table	6.17
table-item	Select Table Item	5.83
queryButton-pane1	Basic Word Search (Visualize a word)	3.67
table-button	Table View Visualization	3.0
queryButton-pane2	Graph Comparison	1.67
pane2-node	Visualization (Select a word in the graph)	1.5
settings-button	Settings Button	1.33
x-button-parallel-coordinates	Remove line from Parallel Coordinates	1.17
year-slider-pane2	Visualize a different year	0.0

Table A.2: Formative Phase: User Story Completion Time Statistics (sorted by Average Duration)

Element Id	**User Story**	**Min Duration (s)**	**Average Duration (s)**	**Max Duration (s)**
year-slider-pane2	Visualize a different year	0	0	0
table-item	Select Table Item	0.38	7.33	52.69
table-sort	Sort Table	0.5	8.78	61.16
pane2-node	Visualization (Select a word in the graph)	1.5	11.72	36.91
year-slider-pane1	Visualize a different year	0.24	14.71	227.56
x-button-parallel-coordinates	Remove line from Parallel Coordinates	0.54	15.28	61.31
settings-button	Settings Button	0.83	16.61	81.22
pane1-node	Visualization (Select a word in the graph)	0.17	19.75	140.49
table-button	Table View Visualization	1.11	22.57	72.84
queryButton-pane2	Graph Comparison	1.77	55.13	175.33
queryButton-pane1	Basic Word Search (Visualize a word)	3.78	72.14	210.48

Table A.3: Formative Phase: User Story Completion Interaction Statistics (sorted by Average Count)

Element Id	**User Story**	**Min Interaction Count**	**Average Interaction Count**	**Max Interaction Count**
year-slider-pane2	Visualize a different year	0	0	0
x-button-parallel-coordinates	Remove line from Parallel Coordinates	1	5.0	9
table-sort	Sort Table	0	5.65	41
table-item	Select Table Item	0	7.34	92
pane2-node	Visualization (Select a word in the graph)	0	7.78	48
pane1-node	Visualization (Select a word in the graph)	0	10.45	98
year-slider-pane1	Visualize a different year	0	11.91	145
table-button	Table View Visualization	1	20.33	95
settings-button	Settings Button	2	30.25	93
queryButton-pane1	Basic Word Search (Visualize a word)	4	66.59	342
queryButton-pane2	Graph Comparison	0	67.0	294

Table A.4: Formative Phase: First Click Duration (from session start) for each element

Element Id	Min Duration	Average Duration	Max Duration
selectSubCorpus-option	2.31	22.79	43.26
info	23.3	47.97	94.06
selectSubCorpus-pane1	1.41	52.18	87.16
selectCorpus-pane1	14.06	59.87	114.82
selectTargetword-pane1	3.0	61.6	122.39
year-option	9.49	75.07	140.65
selectTargetWord-option	6.76	77.88	151.29
zoom-in-button-pane1	59.62	80.11	100.59
selectYear-pane1	6.87	91.81	156.34
queryButton-pane1	26.22	93.49	196.34
search-form-2	129.11	129.11	129.11
second-query-button	51.78	142.11	276.36
zoom-out-button-pane1	53.94	160.38	316.06
pane1-node	107.64	173.02	308.83
settings-button	81.22	198.7	278.53
search-form-1	151.38	240.19	329.01
sidebar	260.59	280.84	301.09
select-all-checkbox-pane1	112.04	281.31	842.86
sidebar-close-button	268.33	285.39	302.46
top-navigation	286.67	286.67	286.67
year-slider-pane1	126.16	295.02	688.88
digits-slider-option	298.45	298.45	298.45
table-sort	130.48	318.44	419.12
selectSubCorpus-pane2	90.46	345.85	964.01
export-csv-button	276.68	380.86	485.04
selectYear-pane2	108.2	410.09	687.15
table-button	117.67	413.51	776.18
show-info-button	412.16	478.97	545.79
selectTargetword-pane2	96.96	502.84	951.96
queryButton-pane2	110.56	528.21	1007.24
x-button-parallel-coordinates	245.85	539.54	833.23
selectCorpus-option	554.4	554.4	554.4
table	120.52	572.99	1090.74
table-item	303.64	606.72	1072.79
select-all-checkbox-pane2	169.59	651.13	1033.06
pane2-node	296.9	688.43	1030.17
font-option	789.66	789.66	789.66
white-label-checkbox-option	795.86	795.86	795.86
bold-checkbox-option	797.94	797.94	797.94
selectCorpus-pane2	992.89	992.89	992.89
selected-words-top-checkbox-option	867.6	1066.6	1265.6

A.2 Pattern Matching

Table A.5: Formative Phase: Mouse Click Repetitions

Element Id	Repetition Count
info	46
selectTargetword-pane1	36
year-slider-pane1	28
table-item	25
pane1-node	23
table-sort	23
selectCorpus-pane1	15
selectYear-pane1	13
zoom-out-button-pane1	10
selectSubCorpus-pane1	10
selectYear-pane2	10
font-option	8
selectTargetword-pane2	8
zoom-in-button-pane1	7
select-all-checkbox-pane1	7
selectSubCorpus-pane2	6

Table A.6: Formative Phase: Repeated Patterns x2

Pattern Pair	Repetition Count
selectTargetword-pane1 -> selectTargetWord-option	18
selectTargetword-pane2 -> selectTargetWord-option	9
selectYear-pane1 -> year-option	7
selectTargetWord-option -> queryButton-pane1	6
year-slider-pane1 -> table-button	6
selectSubCorpus-pane1 -> selectSubCorpus-option	6
table-item -> table-sort	6
select-all-checkbox-pane1 -> pane1-node	5
table-sort -> table-item	5
table-button -> year-slider-pane1	5

Table A.7: Formative Phase: Repeated Patterns x3

Pattern Triple	**Repetition Count**
selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetWord-option	11
selectTargetword-pane1 -> selectTargetWord-option -> queryButton-pane1	5
select-all-checkbox-pane1 -> pane1-node -> pane1-node	5
table-item -> table-sort -> table-sort	5
table-sort -> table-item -> table-item	5

Table A.8: Formative Phase: Repeated Patterns x4

Pattern Quadruple	**Repetition Count**
selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetWord-option	5

Table A.9: Formative Phase: Repeated User Story Patterns x2

Pattern Pair	**Repetition Count**
queryButton-pane1 -> pane1-node	8
table-item -> table-sort	7
table-sort -> table-item	6
year-slider-pane1 -> table-button	6
table-button -> table-sort	5
queryButton-pane1 -> year-slider-pane1	5
table-button -> year-slider-pane1	5

Table A.10: Formative Phase: Repeated User Story Patterns x3

Pattern Triple	**Repetition Count**
table-item -> table-item -> table-sort	6
table-item -> table-sort -> table-sort	6
table-sort -> table-sort -> table-item	6
table-sort -> table-item -> table-item	6
queryButton-pane1 -> pane1-node -> pane1-node	5

Table A.11: Formative Phase: Repeated User Story Patterns x4

Pattern Quadruple	**Repetition Count**
table-sort -> table-sort -> table-item -> table-item	6
table-item -> table-item -> table-sort -> table-sort	5
table-sort -> table-item -> table-item -> table-item	5

A.3 Task based

Table A.12: Formative Phase: Developer Expectations

No	Expectation	Occurrence Count	Non Occurrence Count
1	At most one of the setting options should be changed	4	2
2	First User Event completed is 'visualize a word' (first completion event queryButton-pane1)	5	1
3	Network graph 1 should be zoomed in/out at least once (either through button or through mouse wheel)	4	2
4	Network graph 2 should be zoomed in/out at least once (either through button or through mouse wheel)	2	4
5	Select Subcorpus should be selected before and not after select target word	4	2
6	Select-all-checkbox in the network graph 1 should be clicked at least once	5	1
7	Select-all-checkbox in the network graph 2 should be clicked at least once	5	1
8	Settings clicked not more than 4 times (opened max twice)	5	1
9	User story completed at least once: pane1-node	5	1
10	User story completed at least once: pane2-node	3	3
11	User story completed at least once: queryButton-pane1	6	0
12	User story completed at least once: queryButton-pane2	5	1
13	User story completed at least once: settings-button	3	3
14	User story completed at least once: table-button	6	0
15	User story completed at least once: table-item	3	3
16	User story completed at least once: table-sort	4	2
17	User story completed at least once: x-button-parallel-coordinates	2	4
18	User story completed at least once: year-slider-pane1	4	2
19	User story completed at least once: year-slider-pane2	0	6

A.4 Inferential

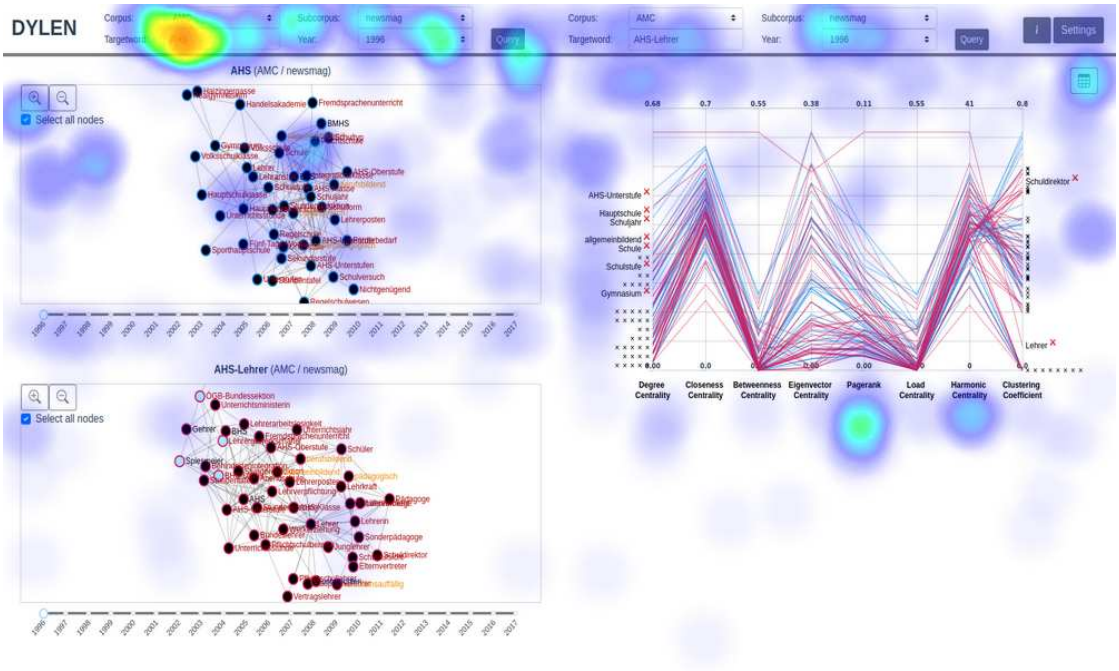

Figure A.1: Formative Phase: Mouse Click Hotspot Map - Screenshot 1

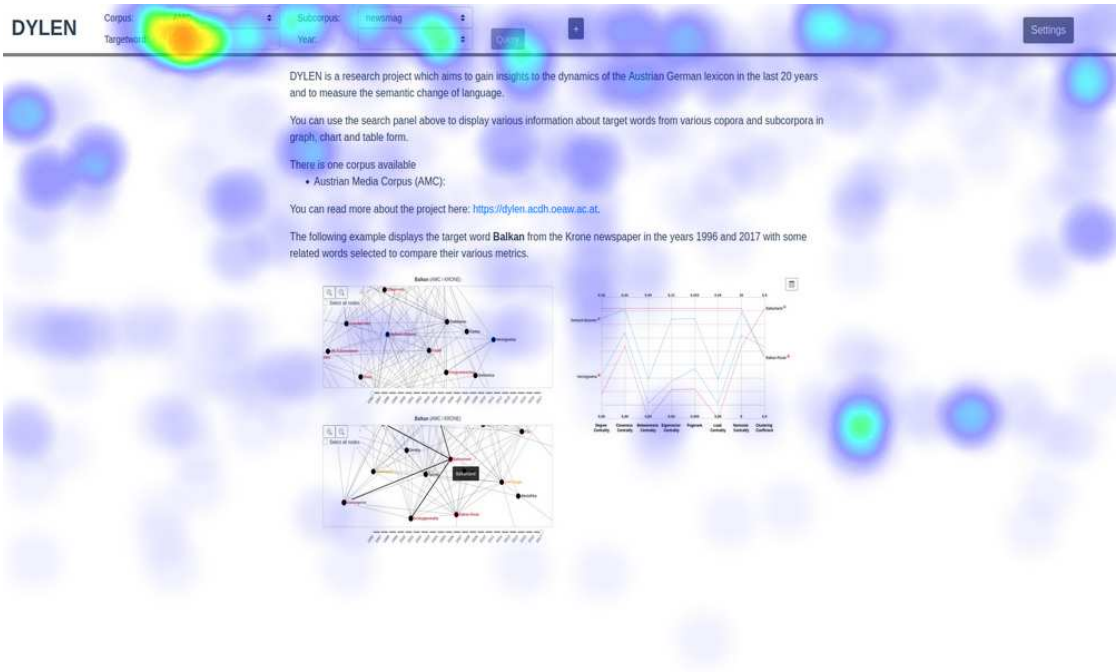

Figure A.2: Formative Phase: Mouse Click Hotspot Map - Screenshot 2

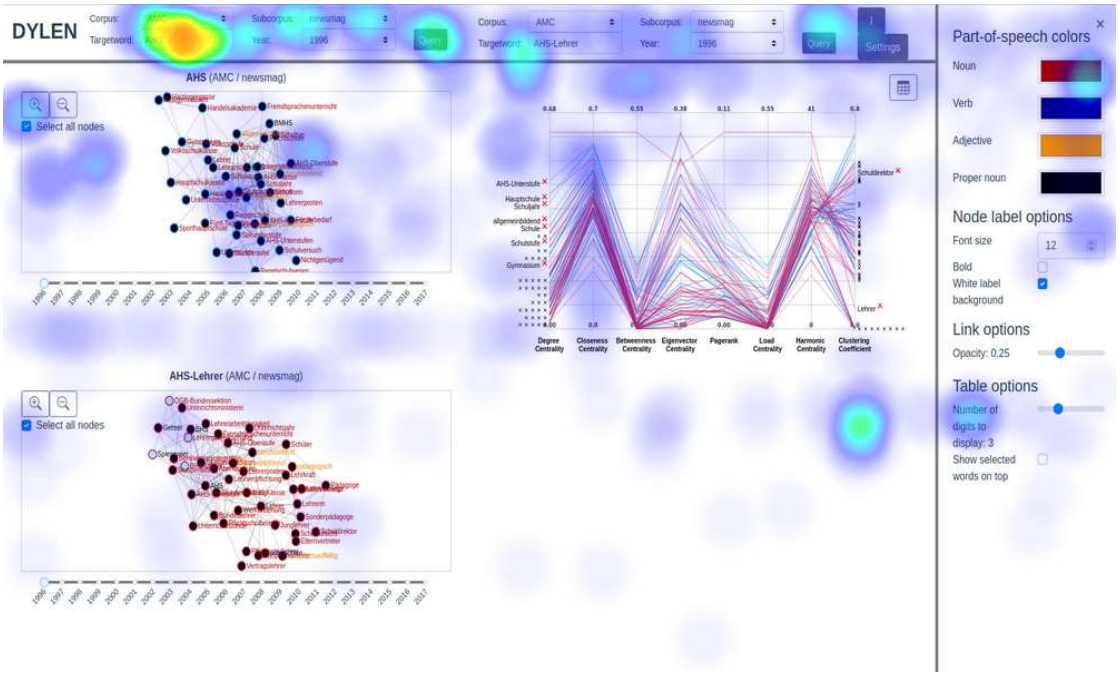

Figure A.3: Formative Phase: Mouse Click Hotspot Map - Screenshot 3

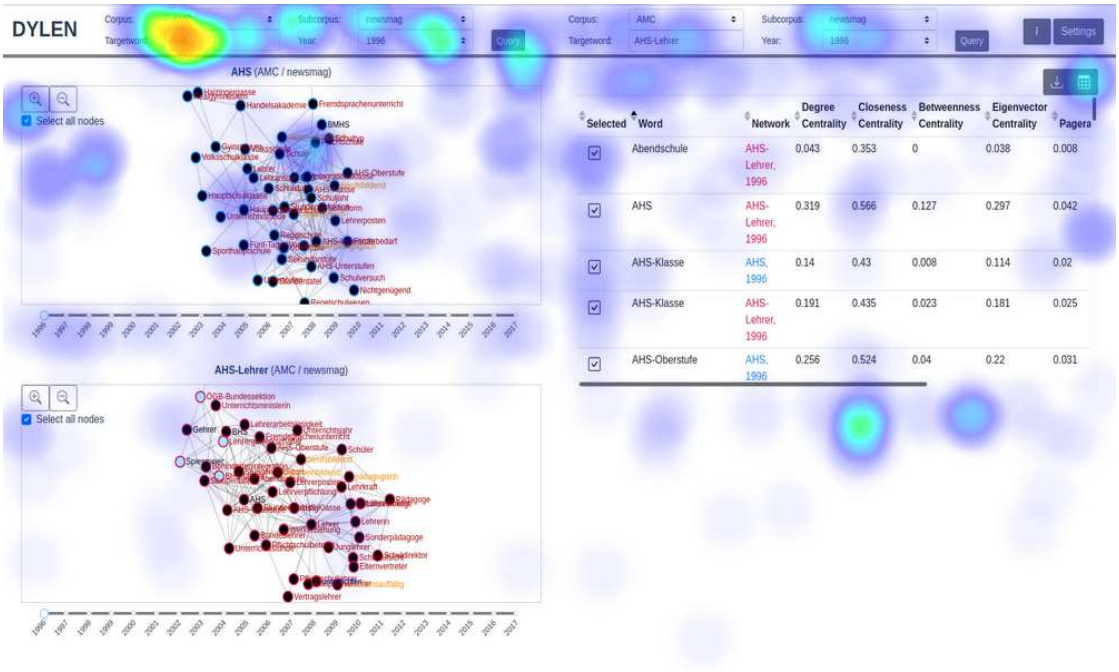

Figure A.4: Formative Phase: Mouse Click Hotspot Map - Screenshot 4

Table A.13: Formative Phase: Click Counts Per Session For Items With More Than 1 Click

Element id	**Click count per Session**
selectTargetword-pane1	10.33
info	8.83
year-slider-pane1	7.33
pane1-node	6.67
table-sort	6.17
table-item	5.83
selectYear-pane1	5.33
selectTargetWord-option	4.83
selectSubCorpus-pane1	4.17
queryButton-pane1	3.67
selectTargetword-pane2	3.5
selectYear-pane2	3.5
selectCorpus-pane1	3.5
select-all-checkbox-pane1	3.17
table-button	3.0
zoom-out-button-pane1	2.5
selectSubCorpus-pane2	2.17
zoom-in-button-pane1	1.67
queryButton-pane2	1.67
search-form-1	1.67
pane2-node	1.5
year-option	1.5
font-option	1.5
settings-button	1.33
search-form-2	1.33
x-button-parallel-coordinates	1.17
select-all-checkbox-pane2	1.17
selectSubCorpus-option	1.17

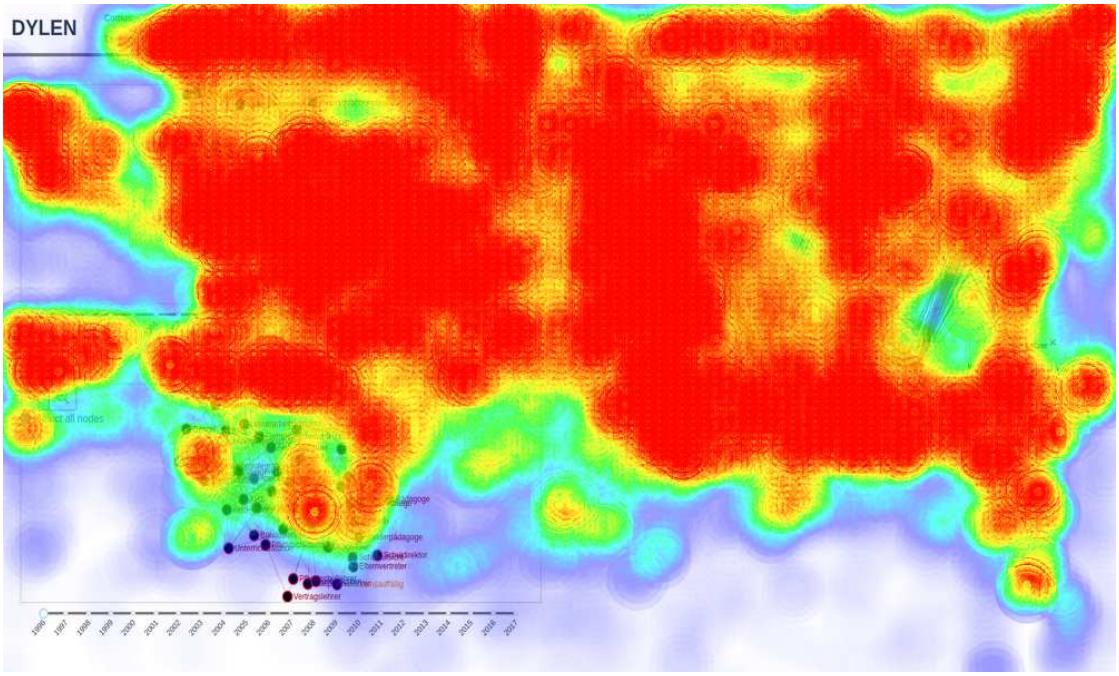

Figure A.5: Formative Phase: Mouse Movement Hotspot Map

Table A.14: Formative Phase: Mouse Over Count Per Session

Element Id	**Mouse Over Count**
results	126.33
search-form-1	48.0
parallel-coordinates	47.0
table	44.0
top-navigation	39.33
search-form-2	37.0
sidebar	21.0
selectYear-pane1	14.83
selectTargetword-pane1	13.83
info	13.0
selectSubCorpus-pane1	11.33
selectYear-pane2	11.0
queryButton-pane1	8.0
selectTargetword-pane2	7.67
selectCorpus-pane1	7.5
table-button	5.67
selectSubCorpus-pane2	5.67

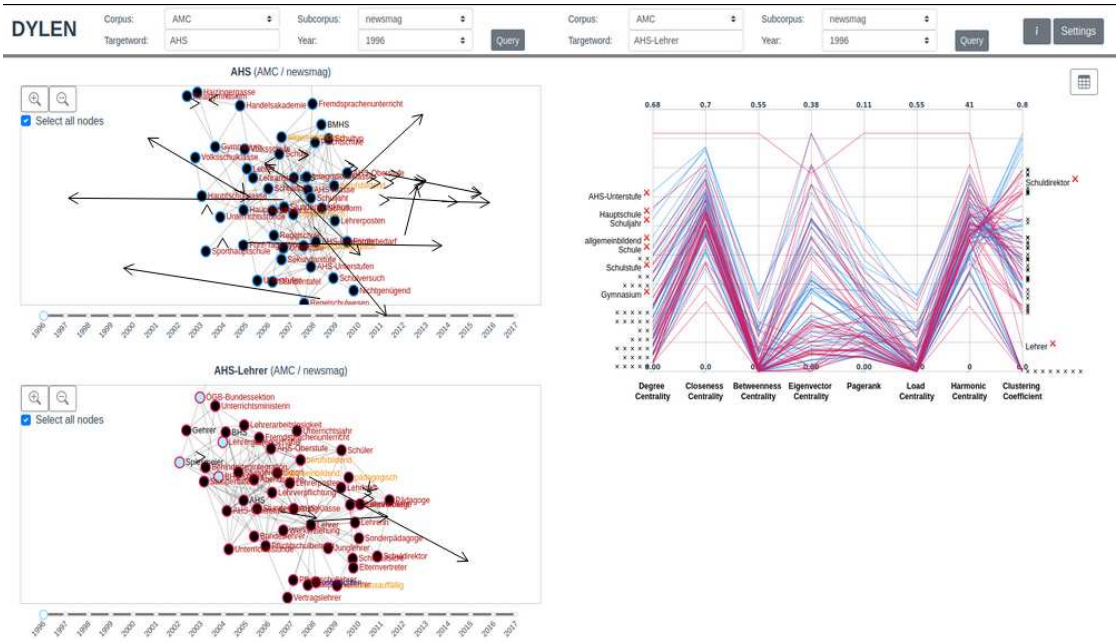

Figure A.6: Formative Phase: Mouse Drag Map

Table A.15: Formative Phase: More Inferential Metrics

Metric	**Min**	**Avg**	**Max**
Scroll Count	0	773.67	1987
Key Press Count	6	34.17	106

APPENDIX B

Appendix: Iterative Phase

B.1 Metric Based

Table B.1: Iterative Phase: First Click Duration (from session start) for each element

Element Id	Min Duration	Average Duration	Max Duration
info-tab-content	9.19	44.98	104.26
info	30.88	51.2	71.52
info-tab	23.35	52.61	85.42
search-form-1	16.59	128.21	263.46
selectTargetword-pane1	25.55	151.24	328.98
selectSubCorpus-pane1	21.79	204.04	471.34
selectTargetWord-option	75.21	213.81	349.23
queryButton-pane1	75.4	237.56	422.21
selectCorpus-pane1	8.1	242.0	519.52
year-slider-pane1	121.03	255.71	379.13
zoom-out-button-pane1	174.05	268.25	362.45
sidebar-close-button	294.22	294.22	294.22
zoom-in-button-pane1	128.04	327.09	603.05
corpusOption	155.24	338.01	520.77
pane1-node	210.56	356.03	624.66
color-option-adjective	379.43	379.43	379.43
subCorpusOption	235.92	380.93	525.95
timeSeries-metric-option	218.15	413.44	763.9
timeSeries-relative-option	219.65	454.96	770.86
pane1-label	250.21	503.11	890.06
parallel-coordinates-x-button	232.99	521.88	681.57
show-info-button	187.59	532.74	1021.32
parallel-coordinates-label	532.87	532.87	532.87
timeSeries-line	422.29	540.4	756.88
parallel-coordinates-selectAll-pane1	558.04	558.04	558.04
parallel-coordinates-deselectAll-pane1	547.81	599.47	651.14
toggleFullScreenButton-pane1	400.45	622.27	844.09
top-navigation	105.72	625.63	1145.54
select-all-checkbox-pane1	622.1	684.21	746.33
settings-button	140.13	689.61	1219.11
selectCorpus-pane2	770.71	770.71	770.71
toggleFullScreenButton-nodeMetrics	404.14	786.24	1213.16
year-slider-pane2	817.09	853.18	889.27
second-query-button	768.24	863.44	974.53
selectTargetword-pane2	776.01	868.74	980.19
queryButton-pane2	779.72	878.2	996.84
pane2-node	828.89	878.27	927.64
table-sort	714.97	881.87	1038.55
selectSubCorpus-pane2	814.28	896.85	982.66
select-all-checkbox-pane2	914.01	914.01	914.01
toggleFullScreenButton-timeSeries	431.48	950.86	1352.48
table	919.89	998.87	1077.85
zoom-in-button-pane2	1051.85	1051.85	1051.85

B.2 Pattern Matching

Table B.2: Iterative Phase: Mouse Click Repetitions

Element Id	Repetition Count
pane1-node	33
zoom-in-button-pane1	32
table-item	29
selectTargetword-pane1	28
info-tab-content	19
timeSeries-metric-option	18
timeSeries-relative-option	15
table-sort	12
selectCorpus-pane1	9
year-slider-pane1	8
parallel-coordinates-x-button	8
table	8
zoom-in-button-pane2	8
info-tab	6
selectSubCorpus-pane1	5

Table B.3: Iterative Phase: Repeated Patterns x2

Pattern Pair	Repetition Count
selectTargetword-pane1 -> selectTargetWord-option	19
selectTargetWord-option -> queryButton-pane1	9
search-form-1 -> selectTargetword-pane1	8
selectCorpus-pane1 -> selectSubCorpus-pane1	7
info-tab -> info-tab-content	7
info-tab-content -> info-tab	7
selectTargetWord-option -> search-form-1	5
selectSubCorpus-pane1 -> selectTargetword-pane1	5
queryButton-pane1 -> selectTargetword-pane1	5

Table B.4: Iterative Phase: Repeated Patterns x3

Pattern Triple	**Repetition Count**
selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetWord-option	10
selectTargetword-pane1 -> selectTargetWord-option -> queryButton-pane1	9
search-form-1 -> selectTargetword-pane1 -> selectTargetword-pane1	5
info-tab -> info-tab-content -> info-tab-content	5
selectTargetword-pane1 -> selectTargetWord-option -> search-form-1	5
selectCorpus-pane1 -> selectCorpus-pane1 -> selectSubCorpus-pane1	5

Table B.5: Iterative Phase: Repeated Patterns x4

Pattern Quadruple	**Repetition Count**
selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetWord-option	5

Table B.6: Iterative Phase: Repeated User Story Patterns x2

Pattern Pair	**Repetition Count**
year-slider-pane1 -> pane1-node	6
queryButton-pane1 -> year-slider-pane1	5

Table B.7: Iterative Phase: Repeated User Story Patterns x3

Pattern Triple	**Repetition Count**
year-slider-pane1 -> pane1-node -> pane1-node	5

B.3 Task based

Table B.8: Iterative Phase: Developer Expectations

Expectation	**Occurrence Count**	**Non Occurrence Count**
At most one of the setting options should be changed	5	1
First User Event completed is “Visualize a word” (first completion event queryButtonpane1)	4	2
Full screen button should be clicked at least once	4	2
Network graph 1 should be zoomed in/out at least once (either through button or through mouse wheel)	5	1
Network graph 2 should be zoomed in/out at least once (either through button or through mouse wheel)	2	4
Nodes in network graph 1 are clicked at least 3 times	3	3
Select Subcorpus should be selected before and not after select target word	2	4
Select all checkbox interaction should be done at least twice(select and deselect) for network 1	3	3
Select all checkbox interaction should be done at least twice(select and deselect) for network 2	2	4
Settings clicked not more than 4 times (opened max twice)	6	0
Table item should be clicked at least 10 times (for selecting words that appear in both networks)	2	4
Table sort should be used at least once	3	3
The resize functionality should be used at least once	6	0
Time series hover at least once	6	0
Time series metric changed at least once	5	1
User should hover over *(overlapping node labels) on parallel coordinates at least twice	6	0
User should hover over parallel coordinates lines at least twice	6	0

Table B.8: Iterative Phase: Developer Expectations - continued

Expectation	**Occurrence Count**	**Non Occurrence Count**
User story completed at least once: pane1-node	5	1
User story completed at least once: pane2-node	2	4
User story completed at least once: parallel-coordinates-x-button	4	2
User story completed at least once: queryButton-pane1	6	0
User story completed at least once: queryButton-pane2	3	3
User story completed at least once: resetQueryButton-pane2	0	6
User story completed at least once: settings-button	4	2
User story completed at least once: table-button	2	4
User story completed at least once: table-item	2	4
User story completed at least once: table-sort	3	3
User story completed at least once: timeSeries-relative-option	4	2
User story completed at least once: year-slider-pane1	4	2
User story completed at least once: year-slider-pane2	2	4

B.4 Inferential

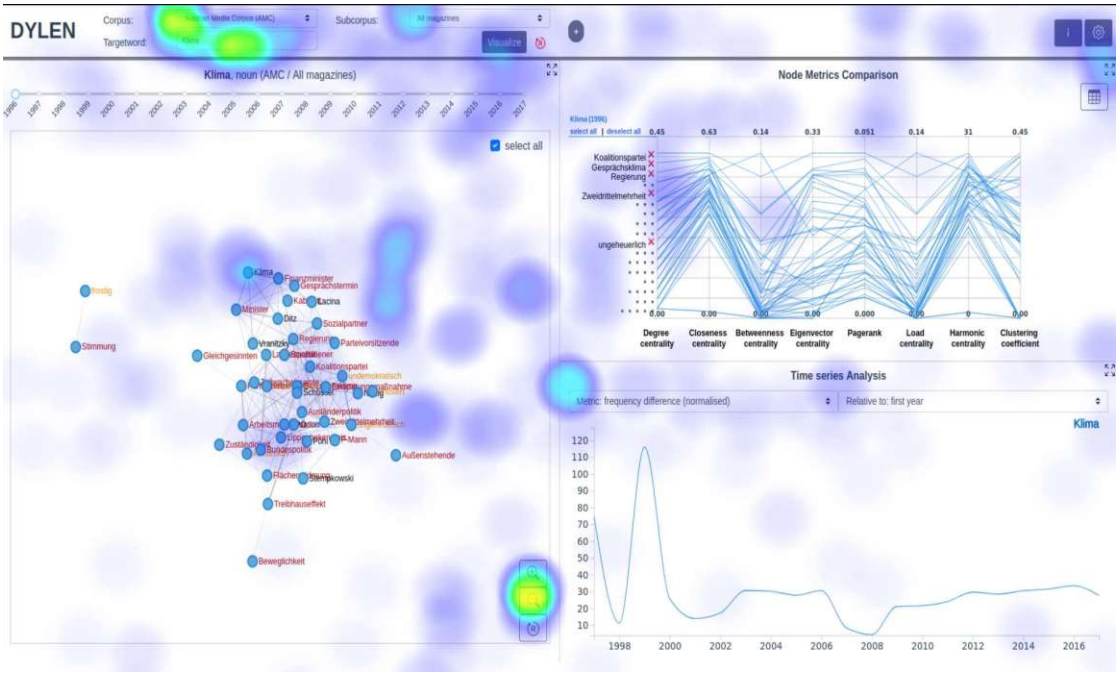

Figure B.1: Iterative Phase: Mouse Click Hotspot Map - Screenshot 1

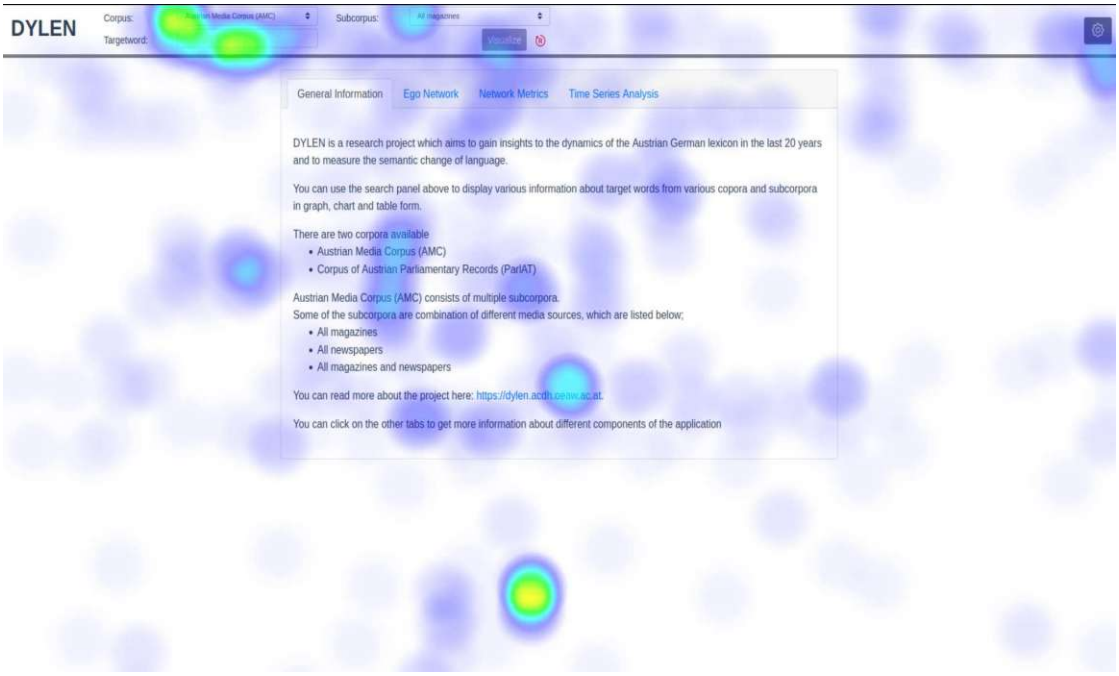

Figure B.2: Iterative Phase: Mouse Click Hotspot Map - Screenshot 2

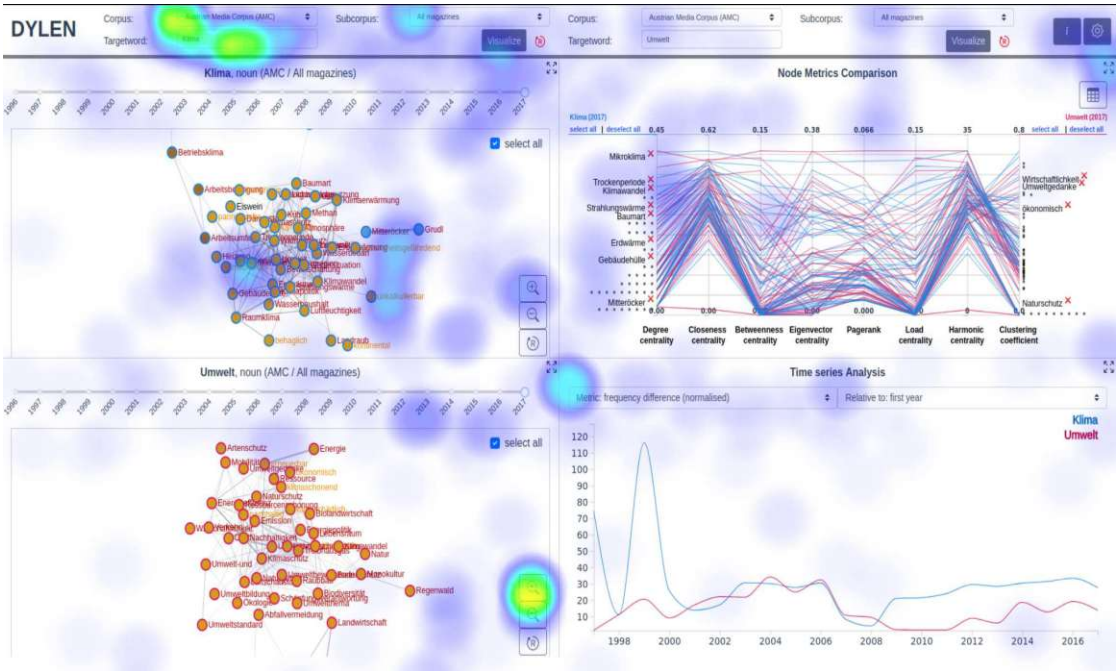

Figure B.3: Iterative Phase: Mouse Click Hotspot Map - Screenshot 3

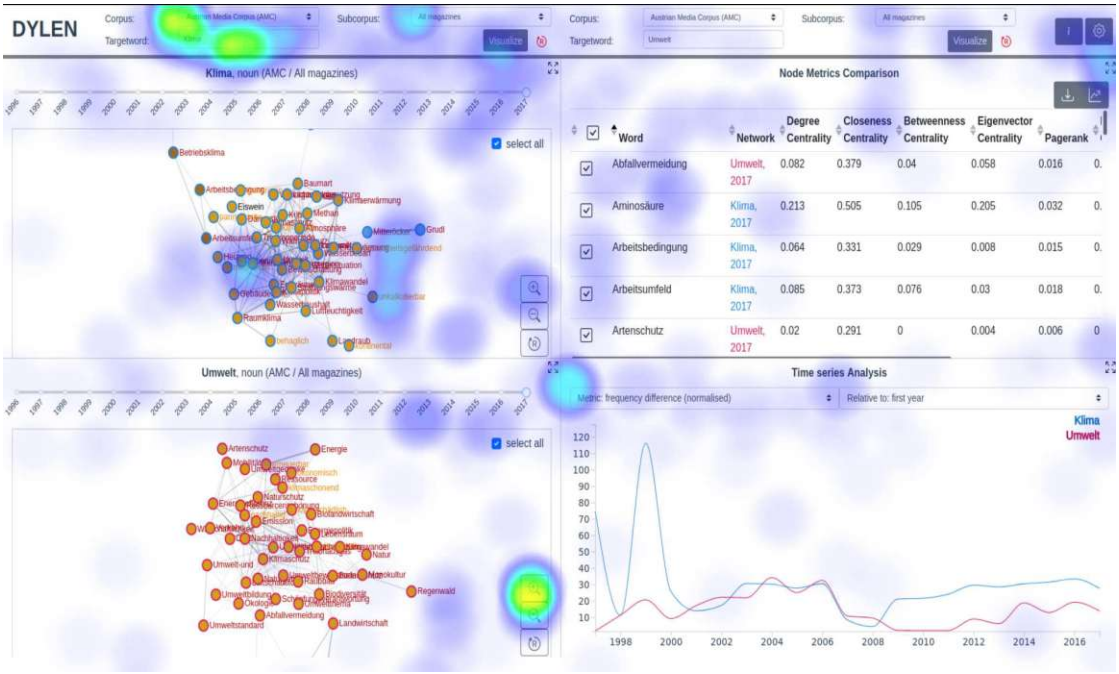

Figure B.4: Iterative Phase: Mouse Click Hotspot Map - Screenshot 4

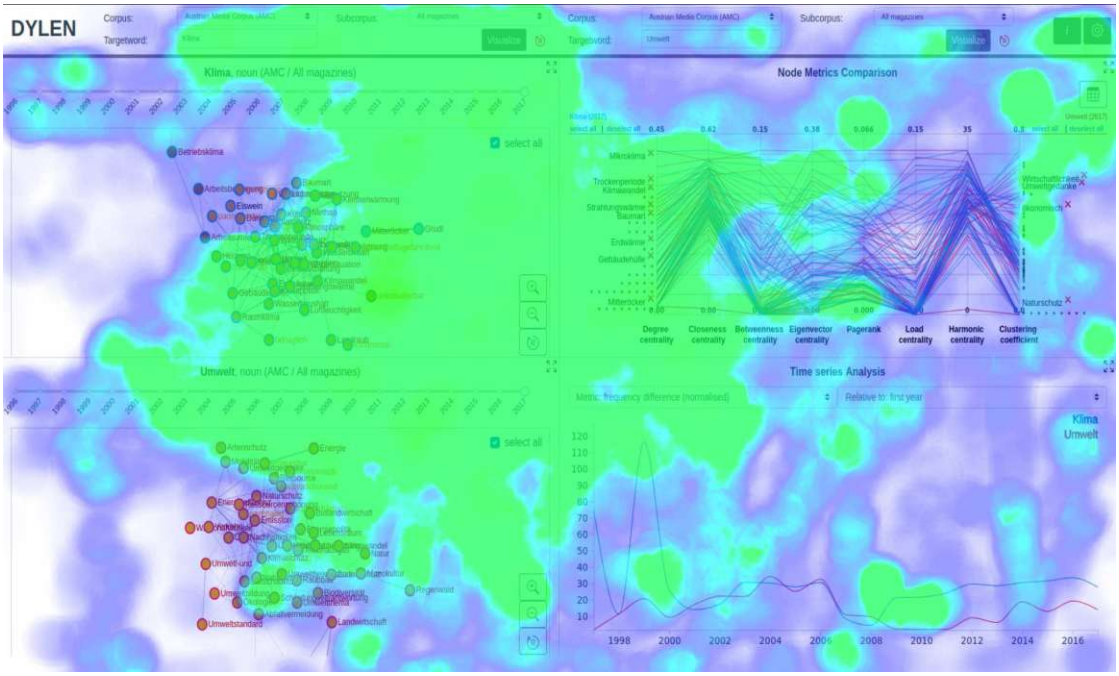

Figure B.5: Iterative Phase: Mouse Movement Hotspot Map

Figure B.6: Iterative Phase: Mouse Drag Map

Table B.9: Iterative Phase: More Inferential Metrics

Metric	**Min**	**Avg**	**Max**
Scroll Count	0	1155.0	5421
Resize Count	1	2.33	6
Key Press Count	5	32.5	41

APPENDIX C

Appendix: Summative Phase

C.1 Metric Based

Table C.1: Summative Phase: User Story Completion Count Per Session

Element Id	**User Story**	**Occurrence Count Per Session**
year-slider-pane1	Visualize a different year	2.11
pane1-node	Visualization (Select a word in the graph)	1.92
queryButton-Ego-pane1	Basic Ego Network Word Search (Visualize a word)	1.41
timeSeries-relative-option	Compare Time Series	1.06
queryButton-Speaker-pane1	Basic General Network Speaker Search (Visualize a speaker)	0.95
queryButton-Party-pane1	Basic General Network Party Search (Visualize a party)	0.93
year-slider-pane2	Visualize a different year	0.46
right-click-pane1-node	View node specific stats (open Context Menu of node)	0.38
node-metrics-table-button	View Node metrics Table Visualization	0.38
queryButton-Ego-pane2	Ego Network Word Graph Comparison	0.33
queryButton-Speaker-pane2	General Network Speaker Graph Comparison	0.26
queryButton-Party-pane2	General Network Party Graph Comparison	0.26
settings-button-nav	Settings Button	0.25
table-sort	Sort Table	0.21
parallel-coordinates-x-button	Remove line from Parallel Coordinates	0.19
pane2-node	Visualization (Select a word in the graph)	0.16
time-series-table-button	View Time series Table Visualization	0.13
settings-button-result	Settings Button	0.13
context-menu-select-as-targetword	Select word from node context menu	0.12
table-item	Select Table Item	0.09
node-metrics-filter	Use filter on node metrics table	0.04
right-click-pane2-node	View node specific stats (open Context Menu of node)	0.04
time-series-filter	Use filter on time series table	0.0

Table C.2: Summative Phase: User Story Completion Time Statistics (sorted by Average Duration)

Element Id	**User Story**	**Min Duration (s)**	**Average Duration (s)**	**Max Duration (s)**
time-series-filter	Use filter on time series table	0	0	0
context-menu-select-as-targetword	Select word from node context menu	0.63	1.15	4.85
node-metrics-filter	Use filter on node metrics table	0.72	3.35	8.34
table-item	Select Table Item	1.0	4.64	18.57
pane2-node	Visualization (Select a word in the graph)	0.18	5.87	35.97
timeSeries-relative-option	Compare Time Series	0.04	9.2	85.53
time-series-table-button	View Time series Table Visualization	0.71	10.47	40.22
right-click-pane1-node	View node specific stats (open Context Menu of node)	0.58	11.29	98.27
right-click-pane2-node	View node specific stats (open Context Menu of node)	4.49	11.62	32.34
year-slider-pane2	Visualize a different year	0.28	17.15	211.07
settings-button-result	Settings Button	1.33	19.51	68.37

Table C.2: Summative Phase: User Story Completion Time Statistics (sorted by Average Duration) - continued

Element Id	**User Story**	**Min Duration (s)**	**Average Duration (s)**	**Max Duration (s)**
pane1-node	Visualization (Select a word in the graph)	0.1	21.59	721.62
queryButton-Party-pane2	General Network Party Graph Comparison	1.16	22.58	95.68
year-slider-pane1	Visualize a different year	0.92	23.64	1022.57
queryButton-Ego-pane2	Ego Network Word Graph Comparison	1.97	24.11	180.49
table-sort	Sort Table	0.61	24.7	285.92
parallel-coordinates-x-button	Remove line from Parallel Coordinates	0.27	37.34	526.72
node-metrics-table-button	View Node metrics Table Visualization	1.99	46.85	488.76
queryButton-Party-pane1	Basic General Network Party Search (Visualize a party)	2.65	54.54	836.94
queryButton-Speaker-pane1	Basic General Network Speaker Search (Visualize a speaker)	0.92	70.74	1282.34
queryButton-Speaker-pane2	General Network Speaker Graph Comparison	1.91	79.13	787.76
queryButton-Ego-pane1	Basic Ego Network Word Search (Visualize a word)	0.06	107.88	3224.78
settings-button-nav	Settings Button	0.85	119.06	1443.66

Table C.3: Summative Phase: User Story Completion Interaction Statistics (sorted by Average Count)

Element Id	**User Story**	**Min Interaction Count**	**Average Interaction Count**	**Max Interaction Count**
time-series-filter	Use filter on time series table	0	0	0
context-menu-select-as-targetword	Select word from node context menu	0	0.0	0
node-metrics-filter	Use filter on node metrics table	0	5.0	15
right-click-pane1-node	View node specific stats (open Context Menu of node)	0	6.51	28
right-click-pane2-node	View node specific stats (open Context Menu of node)	0	6.6	13
pane2-node	Visualization (Select a word in the graph)	0	6.72	54
table-item	Select Table Item	1	8.1	25
pane1-node	Visualization (Select a word in the graph)	0	9.08	132
table-sort	Sort Table	0	11.33	119
timeSeries-relative-option	Compare Time Series	0	11.61	86
year-slider-pane1	Visualize a different year	0	11.62	154

Table C.4: Summative Phase: User Story Completion Interaction Statistics (sorted by Average Count) - continued

Element Id	**User Story**	**Min Interaction Count**	**Average Interaction Count**	**Max Interaction Count**
time-series-table-button	View Time series Table Visualization	0	11.87	38
year-slider-pane2	Visualize a different year	0	12.53	72
parallel-coordinates-x-button	Remove line from Parallel Coordinates	0	20.43	159
settings-button-nav	Settings Button	0	24.89	175
settings-button-result	Settings Button	4	25.33	68
queryButton-Party-pane2	General Network Party Graph Comparison	2	33.48	125
node-metrics-table-button	View Node metrics Table Visualization	0	35.17	120
queryButton-Ego-pane2	Ego Network Word Graph Comparison	5	35.51	115
queryButton-Speaker-pane1	Basic General Network Speaker Search (Visualize a speaker)	0	36.63	137
queryButton-Party-pane1	Basic General Network Party Search (Visualize a party)	6	48.39	345
queryButton-Speaker-pane2	General Network Speaker Graph Comparison	4	49.28	242
queryButton-Ego-pane1	Basic Ego Network Word Search (Visualize a word)	0	74.17	556

Table C.5: Summative Phase: First Click Duration (from session start) for each element

Element Id	Min Duration	Average Duration	Max Duration
info	1.26	75.48	1467.91
info-tab	0.98	84.82	1465.38
info-tab-content	3.19	124.33	668.68
ego-network-tab	1.45	153.23	3191.06
left-query-bar	63.32	156.1	229.91
info-collapsable-button	3.21	163.95	3169.44
selectGeneralMetric-pane2	96.15	183.78	312.1
selectTargetword-pane1	2.82	185.28	3205.74
font-option	61.35	187.94	314.53
table-select-all	204.09	204.09	204.09
selectTargetWord-option	6.09	213.81	3223.52
queryButton-Ego-pane1	7.3	216.61	3224.78
general-network-tab	1.38	222.36	2864.03
context-menu-select-as-targetword	26.01	227.16	652.89
selectTargetword-pane2	9.15	229.0	1053.05
resetQueryButton-Ego-pane1	11.93	231.55	541.9
selectCorpus-pane2	59.28	231.65	551.15
selectSubCorpus-pane1	8.17	237.75	3201.64
selectCorpus-pane1	8.76	243.8	3198.06
queryButton-Party-pane1	2.65	250.34	2868.34
selectGeneralMetric-pane1	11.28	251.33	1450.53
color-option-verb	81.52	253.91	565.15
zoom-reset-button-pane2	231.46	255.72	279.98
zoom-out-button-pane1	90.92	264.3	412.45
node-filter-sliderpane1	5.38	266.25	1164.28
select-all-checkbox-pane2	171.91	272.78	373.65
zoom-in-button-pane1	99.66	278.25	515.23
digits-slider-option	27.57	281.96	608.07
node-metrics-table-button	11.36	284.15	810.39
resetQueryButton-Speaker-pane2	151.97	284.22	434.29
pane2-label	140.7	287.04	378.05
sidebar	23.21	291.94	755.85
parallel-coordinates-deselectAll-pane2	156.68	292.05	369.5
search-form-1	12.44	296.87	2866.61
selectParty-pane1	2.53	301.48	2202.85
year-slider-pane1	12.22	303.23	1299.35
color-option-proper_noun	305.26	305.26	305.26
table-sort	20.7	305.61	834.35
table	119.72	308.34	515.53

Table C.5: Summative Phase: First Click Duration (from session start) for each element - continued

Element Id	Min Duration	Average Duration	Max Duration
pane1-node	33.06	308.48	1654.87
node-metrics-filter	103.8	312.39	830.88
color-option-adjective	133.6	317.83	637.16
bold-checkbox-option	318.44	318.44	318.44
queryButton-Ego-pane2	12.8	325.42	1566.45
pane2-node	142.37	334.97	798.93
select-all-checkbox-pane1	101.49	339.02	1252.26
show-clusters-checkbox-pane1	102.88	342.56	1148.12
node-filter-sliderpane2	125.68	346.33	750.82
parallel-coordinates-x-button	56.18	347.96	1107.46
timeSeries-relative-option	42.78	354.5	1240.94
queryButton-Party-pane2	46.16	362.65	1905.8
parallel-coordinates-selectAll-pane2	366.77	366.77	366.77
table-item	102.73	374.07	822.53
general-network-speaker-tab	5.72	390.26	2542.72
toggleFullScreenButton-nodeMetrics	152.34	390.3	756.58
selectSubCorpus-pane2	48.63	397.87	1050.44
timeSeries-metric-option	72.07	399.87	1225.06
selectSpeaker-pane1	7.79	403.24	2549.84
search-form-2	51.94	413.0	933.54
queryButton-Speaker-pane1	13.18	455.21	2579.02
resetQueryButton-Speaker-pane1	150.96	457.42	1067.13
toggleFullScreenButton-pane1	19.01	462.15	1961.14
subCorpusOption	178.38	463.44	1012.09
right-click-pane1-node	24.92	467.61	1659.83
selectParty-pane2	44.43	468.3	2089.15
pane1-label	63.04	472.65	1388.79
resetQueryButton-Ego-pane2	244.14	473.8	841.98
timeSeries-line	109.23	478.48	1651.32
year-slider-pane2	17.71	480.29	2109.64
selected-words-top-checkbox-option	214.54	488.76	762.99
partyOption	3.45	496.23	1924.08
settings-button-result	21.6	499.03	1892.32
parallel-coordinates-deselectAll-pane1	158.04	500.19	961.56
sidebar-close-button	9.93	501.95	1898.17
parallel-coordinates-label	137.44	517.46	1108.66
top-navigation	0.86	518.86	3184.71
minimum-similarity-slider-option	167.58	521.18	1320.57
toggleFullScreenButton-pane2	31.45	531.01	1969.13

Table C.5: Summative Phase: First Click Duration (from session start) for each element - continued

Element Id	Min Duration	Average Duration	Max Duration
metricGeneralOption	144.42	537.57	1875.88
resetQueryButton-Party-pane1	38.06	544.55	1522.77
parallel-coordinates-selectAll-pane1	182.76	546.23	959.31
time-series-table-button	57.67	568.54	1267.84
show-info-button	79.82	604.96	2068.77
settings-button-nav	1.38	645.19	3286.8
color-option-noun	95.42	658.61	2701.57
queryButton-Speaker-pane2	117.48	667.62	2652.58
toggleFullScreenButton-timeSeries	273.2	670.35	1265.42
resetQueryButton-Party-pane2	116.63	680.13	1523.22
bottom-info-bar	1.73	682.98	1639.82
dysen-link	27.06	687.52	1728.14
time-series-info-button	49.1	699.23	1197.17
corpusOption	30.42	719.55	3200.32
parallel-coordinates-axis-checkbox-option	78.19	726.55	2711.46
selectSpeaker-pane2	110.68	755.55	2621.92
node-filter-info	651.8	776.37	900.94
speakerOption	75.46	787.41	2578.07
zoom-out-button-pane2	131.56	853.9	2151.68
parallel-coordinates-info-button	614.26	867.24	1122.2
zoom-reset-button-pane1	230.15	871.66	1240.88
timeSeries	369.02	917.07	1652.52
zoom-in-button-pane2	281.34	918.57	2150.49
opacity-slider-option	156.66	948.41	3291.06
right-click-pane2-node	412.36	1010.54	1670.96
ego-network-info-button	149.1	1174.3	2583.52
show-clusters-checkbox-pane2	1241.93	1241.93	1241.93

C.2 Pattern Matching

Table C.6: Summative Phase: Mouse Click Repetitions

Element Id	Repetition Count
info-tab	234
info-tab-content	157
info-collapsable-button	155
zoom-in-button-pane1	140
year-slider-pane1	114
pane1-node	111
timeSeries-relative-option	79
selectTargetword-pane1	74
info	73
selectParty-pane1	63
timeSeries-metric-option	60

Table C.7: Summative Phase: Repeated Patterns x2

Pattern Pair	Repetition Count
selectTargetWord-option -> queryButton-Ego-pane1	135
info-tab -> info-collapsable-button	128
selectTargetword-pane1 -> selectTargetWord-option	119
info-collapsable-button -> info-tab	80
info-collapsable-button -> info-tab-content	61
info-tab-content -> info-collapsable-button	51

Table C.8: Summative Phase: Repeated Patterns x3

Pattern Triple	Repetition Count
selectTargetword-pane1 -> selectTargetWord-option -> queryButton-Ego-pane1	113
info-tab -> info-collapsable-button -> info-collapsable-button	69
info-tab -> info-tab -> info-collapsable-button	49
info-collapsable-button -> info-collapsable-button -> info-tab	43
info-collapsable-button -> info-tab -> info-collapsable-button	36
selectTargetword-pane2 -> selectTargetWord-option -> queryButton-Ego-pane2	31

Table C.9: Summative Phase: Repeated Patterns x4

Pattern Quadruple	**Repetition Count**
info-tab -> info-collapsable-button -> info-collapsable-button -> info-collapsable-button	35
info-tab -> info-tab -> info-collapsable-button -> info-collapsable-button	29
info-collapsable-button -> info-collapsable-button -> info-collapsable-button -> info-tab	26
selectTargetword-pane1 -> selectTargetword-pane1 -> selectTargetWord-option -> queryButton-Ego-pane1	25
ego-network-tab -> selectTargetword-pane1 -> selectTargetWord-option -> queryButton-Ego-pane1	25
selectSubCorpus-pane1 -> selectTargetword-pane1 -> selectTargetWord-option -> queryButton-Ego-pane1	23

Table C.10: Summative Phase: Repeated User Story Patterns x2

Pattern Pair	**Repetition Count**
queryButton-Ego-pane1 -> year-slider-pane1	38
year-slider-pane1 -> pane1-node	21
queryButton-Ego-pane1 -> queryButton-Ego-pane2	21
queryButton-Ego-pane1 -> pane1-node	19
queryButton-Party-pane1 -> year-slider-pane1	18
queryButton-Speaker-pane1 -> year-slider-pane1	16
pane1-node -> queryButton-Ego-pane1	16
year-slider-pane1 -> queryButton-Ego-pane1	16

Table C.11: Summative Phase: Repeated User Story Patterns x3

Pattern Triple	**Repetition Count**
queryButton-Ego-pane1 -> year-slider-pane1 -> year-slider-pane1	26
queryButton-Ego-pane1 -> pane1-node -> pane1-node	10
year-slider-pane1 -> year-slider-pane1 -> pane1-node	10

Table C.12: Summative Phase: Repeated User Story Patterns x4

Pattern Quadruple	**Repetition Count**
queryButton-Ego-pane1 -> year-slider-pane1 -> year-slider-pane1 -> year-slider-pane1	18

C.3 Task based

Table C.13: Summative Phase: Developer Expectations

No	Expectation	Occurrence Count	Non Occurrence Count
1	A network graph should be zoomed in/out at least once (either through button or through mouse wheel)	67	45
2	At least 15% of the users change at least one parallel coordinates axis from settings	6	106
3	At least 20% of users click on the Dysen link	5	107
4	At least 30% of the users change general network query metric or percentage of nodes at least once	7	105
5	At least 5% of the users download table as json/csv at least once	0	112
6	At least 50% of the users change general network query party or speaker at least once	16	96
7	At most one of the setting options should be changed	102	10
8	First User Event completed is on of the “visualize a word/party/speaker” (queryButton-Ego-pane1/queryButton-Party-pane1/queryButton-Speaker-pane1)	88	24
9	Full screen button should be clicked at least once	29	83
10	Info tabs clicked at least once	85	27
11	One of the reset query buttons is clicked at least once	23	89
12	Select Subcorpus should be selected before and not after select target word	60	52
13	Select all interaction(all select all/deselect all buttons) should be done at least once	25	87
14	Settings buttons clicked not more than 6 times	93	19
15	Show clusters interaction should be done at least once	22	90

Table C.13: Summative Phase: Developer Expectations - continued

No	Expectation	Occurrence Count	Non Occurrence Count
16	The resize functionality should be used at least once	92	20
17	Time series hover at least once	68	44
18	Time series metric changed at least once	26	86
19	User gets general network query timeout at most once	109	3
20	User should hover over *(overlapping node labels) on parallel coordinates at least once	31	81
21	User should hover over parallel coordinates lines at least once	56	56
22	User story completed at least once: context-menu-select-as-targetword	5	107
23	User story completed at least once: node-metrics-filter	4	108
24	User story completed at least once: node-metrics-table-button	25	87
25	User story completed at least once: pane1-node	40	72
26	User story completed at least once: pane2-node	7	105
27	User story completed at least once: parallel-coordinates-x-button	7	105
28	User story completed at least once: queryButton-Ego-pane1	65	47
29	User story completed at least once: queryButton-Ego-pane2	29	83
30	User story completed at least once: queryButton-Party-pane1	53	59

Table C.13: Summative Phase: Developer Expectations - continued

No	Expectation	Occurrence Count	Non Occurrence Count
31	User story completed at least once: queryButton-Party-pane2	20	92
32	User story completed at least once: queryButton-Speaker-pane1	42	70
33	User story completed at least once: queryButton-Speaker-pane2	15	97
34	User story completed at least once: right-click-pane1-node	10	102
35	User story completed at least once: right-click-pane2-node	3	109
36	User story completed at least once: settings-button-nav	17	95
37	User story completed at least once: settings-button-result	12	100
38	User story completed at least once: table-item	6	106
39	User story completed at least once: table-sort	10	102
40	User story completed at least once: time-series-filter	0	112
41	User story completed at least once: time-series-table-button	9	103
42	User story completed at least once: timeSeries-relative-option	27	85
43	User story completed at least once: year-slider-pane1	57	55
44	User story completed at least once: year-slider-pane2	20	92

C.4 Inferential

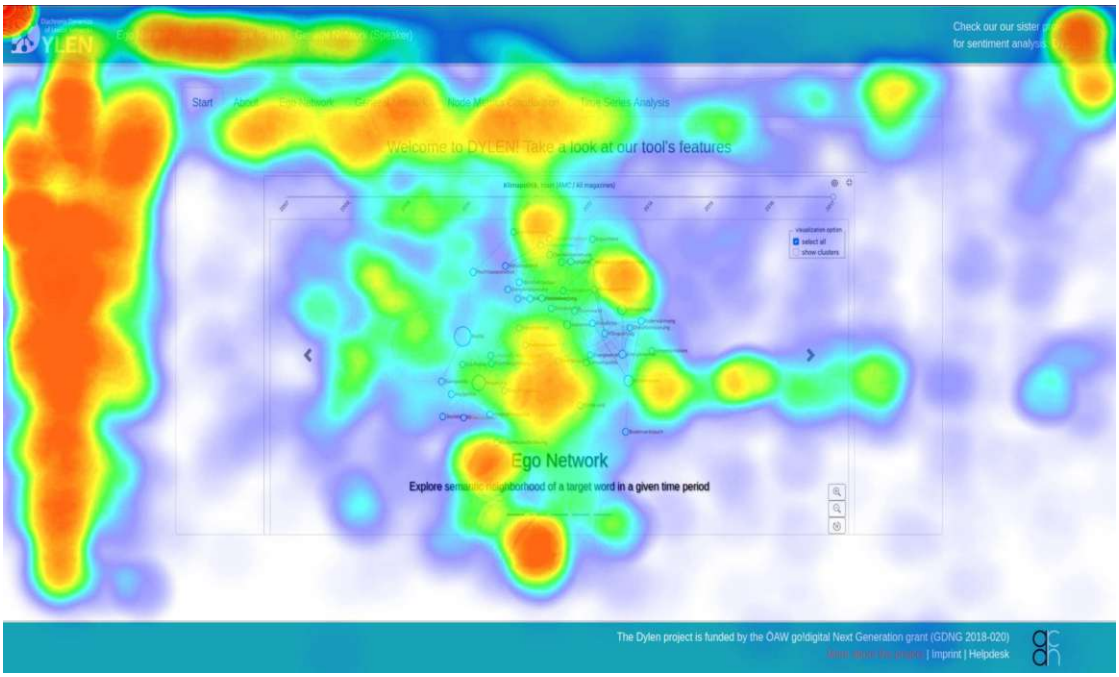

Figure C.1: Summative Phase: Mouse Click Hotspot Map - Screenshot 1

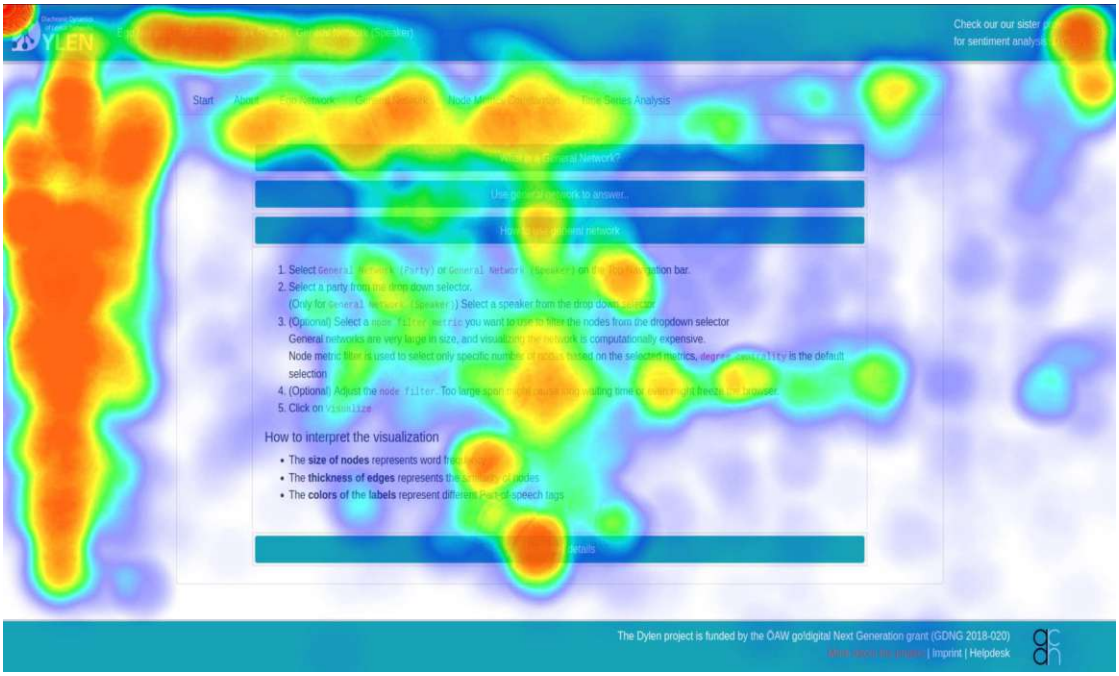

Figure C.2: Summative Phase: Mouse Click Hotspot Map - Screenshot 2

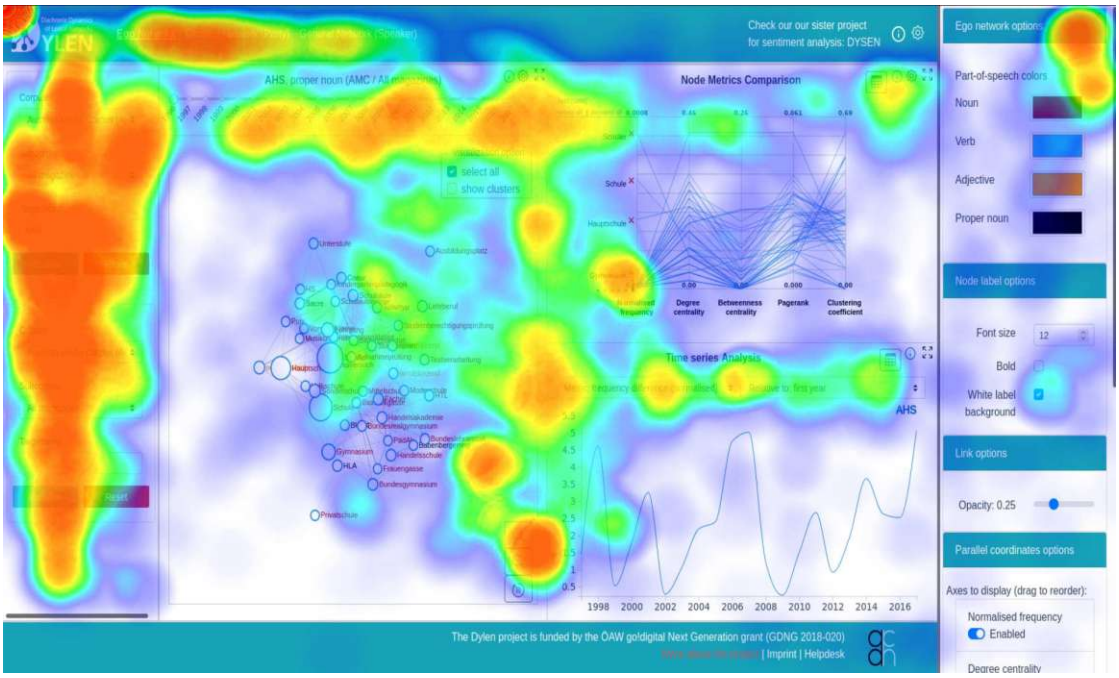

Figure C.3: Summative Phase: Mouse Click Hotspot Map - Screenshot 3

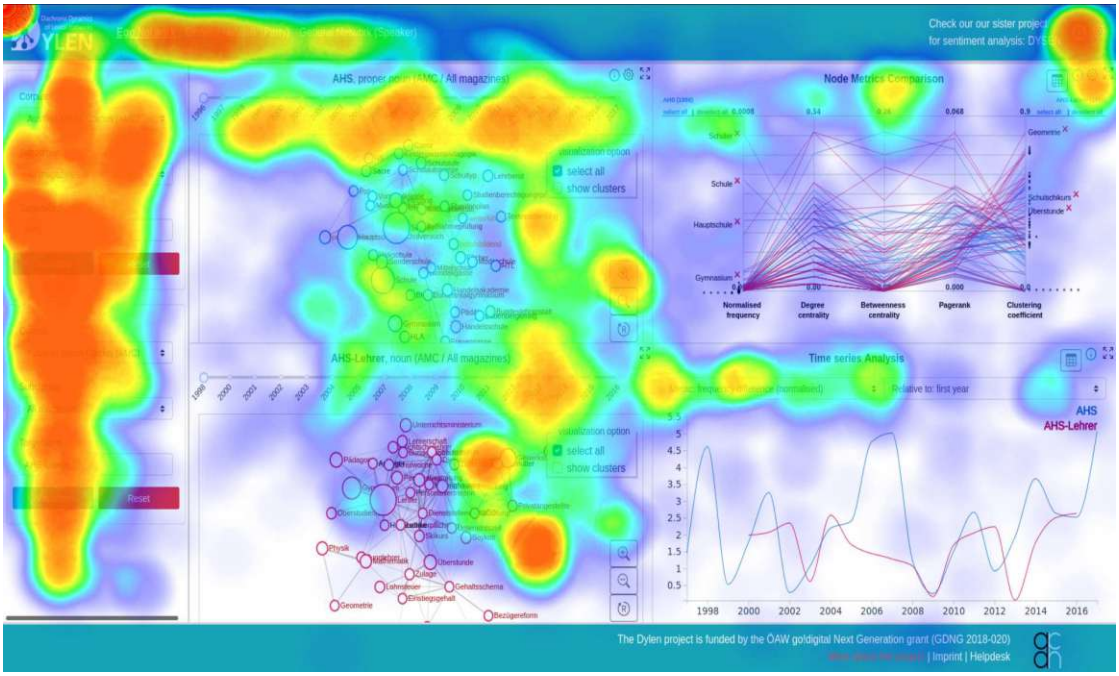

Figure C.4: Summative Phase: Mouse Click Hotspot Map - Screenshot 4

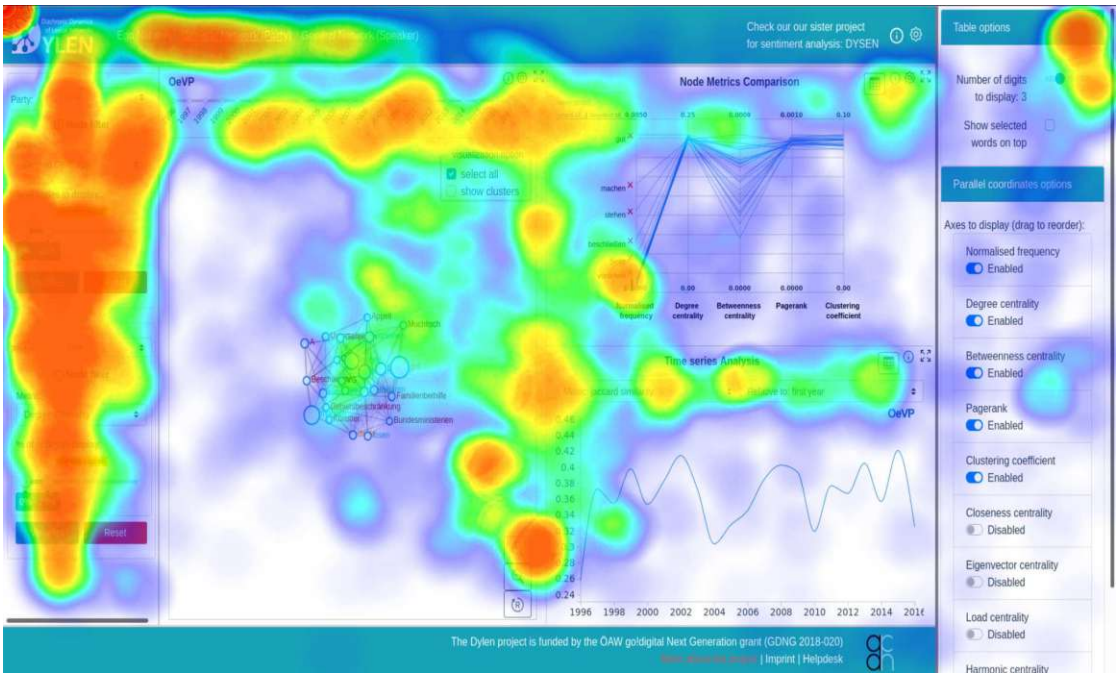

Figure C.5: Summative Phase: Mouse Click Hotspot Map - Screenshot 5

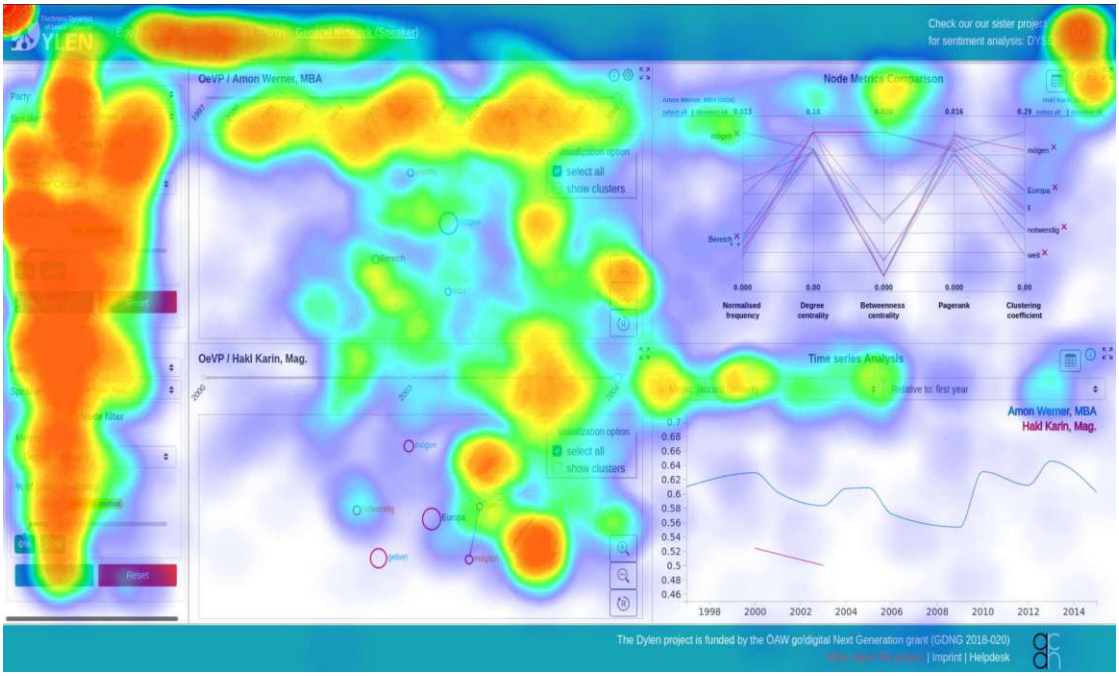

Figure C.6: Summative Phase: Mouse Click Hotspot Map - Screenshot 6

Table C.14: Summative Phase: Click Counts Per Session For Items With More Than 1 Click

Element id	**Click count per Session**
info-tab	4.3
info-collapsable-button	3.02
info-tab-content	2.28
selectTargetword-pane1	2.27
year-slider-pane1	2.11
pane1-node	1.92
selectTargetWord-option	1.59
zoom-in-button-pane1	1.46
queryButton-Ego-pane1	1.41
selectParty-pane1	1.38
info	1.2
ego-network-tab	1.14
timeSeries-relative-option	1.06
selectSpeaker-pane1	1.02

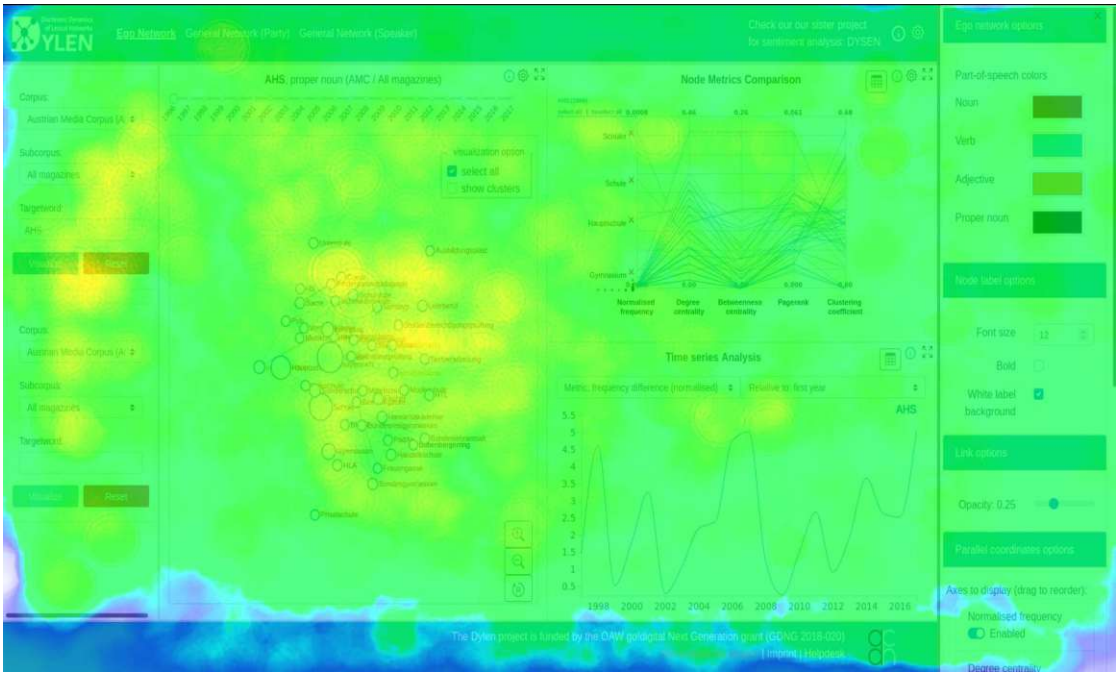

Figure C.7: Summative Phase: Mouse Movement Hotspot Map

Table C.15: Summative Phase: Mouse Over Count Per Session

Element Id	**Mouse Over Count**
search-form-1	33.05
results	30.71
info-tab-content	19.24
top-navigation	17.47
parallel-coordinates	17.4
info-collapsable-button	15.37
search-form-2	11.76
left-query-bar	11.01
info	10.27
parallel-coordinates-line	10.01

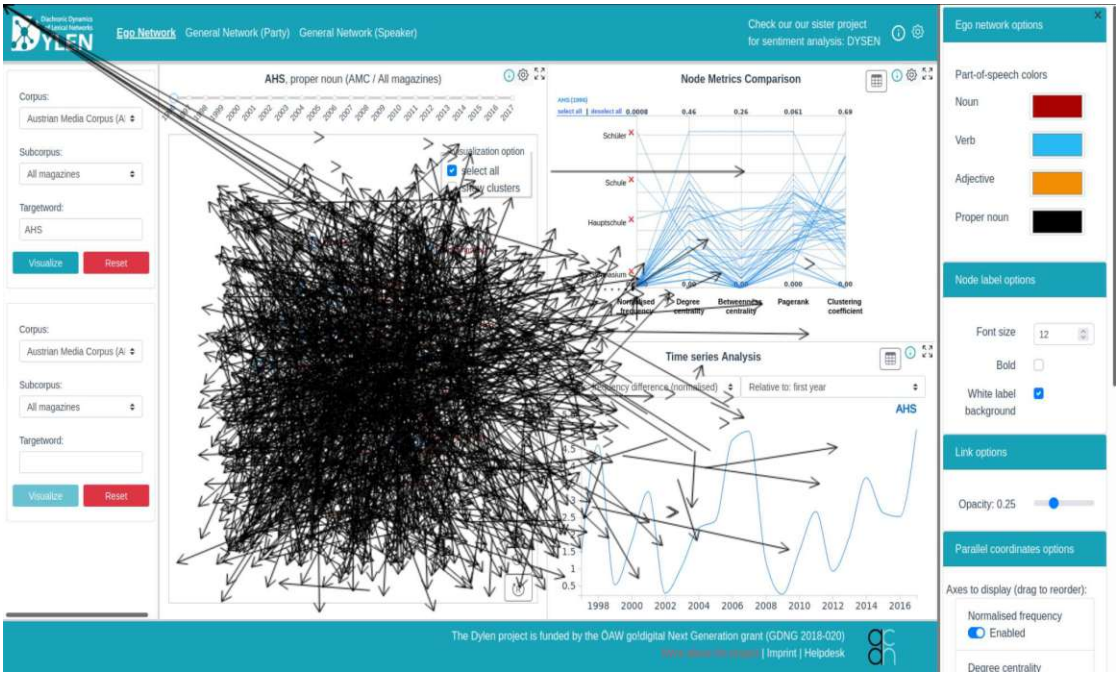

Figure C.8: Summative Phase: Mouse Drag Map

Table C.16: Summative Phase: Durations Per Page

Page	Duration per Session(s)	Percentage
info	289.6	37.12%
ego-network-tab	212.93	27.29%
general-network-tab	146.73	18.81%
general-network-speaker-tab	131.01	16.79%

Table C.17: Summative Phase: Clicks Per Page

Page	Click Count per Session(s)	Percentage
ego-network-tab	19.83	42.88%
info	9.88	21.35%
general-network-tab	9.04	19.54%
general-network-speaker-tab	7.51	16.24%

Table C.18: Summative Phase: More Inferential Metrics

Metric	**Min**	**Avg**	**Max**
Scroll Count	0	284.04	4728
Resize Count	0	5.7	31
Key Press Count	0	12.88	84
Timeout Count	0	0.15	2

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] I Kahfie, MC Ramadan, Salahudin Rafi, and D Perawati. The crash of boeing 737 max 8 and it's effect on costumer trust: Case on lion air passenger. *Advances in Transportation and Logistics Research*, 2:764–769, 2019.

[2] Sara Baase, Michael Nagenborg, Anthony Brinkman, Robert Boyd Skipper, and Becky Cox-White. A gift of fire: Social, legal, and ethical issues for computing and the internet. In *Frontiers in Education Conference, 1997. 27th Annual Conference.'Teaching and Learning in an Era of Change'. Proceedings.*, volume 20, 2008.

[3] Carol M Barnum. *Usability testing essentials: ready, set... test!* Morgan Kaufmann, 2020.

[4] Melody Y Ivory and Marti A Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516, 2001.

[5] Simon Baker, Fiora Au, Gillian Dobbie, and Ian Warren. Automated usability testing using hui analyzer. In *19th Australian Conference on Software Engineering (aswec 2008)*, pages 579–588. IEEE, 2008.

[6] Fiora TW Au, Simon Baker, Ian Warren, and Gillian Dobbie. Automated usability testing framework. In *Proceedings of the ninth conference on Australasian user interface-Volume 76*, pages 55–64, 2008.

[7] Wolfgang Kluth, Karl-Heinz Krempels, and Christian Samsel. Automated usability testing for mobile applications. In *WEBIST (2)*, pages 149–156, 2014.

[8] Florian Lettner and Clemens Holzmann. Usability evaluation framework. In *International Conference on Computer Aided Systems Theory*, pages 560–567. Springer, 2011.

[9] Miroslav Bures, Miroslav Macik, Bestoun S Ahmed, Vaclav Rechtberger, and Pavel Slavik. Testing the usability and accessibility of smart tv applications using an automated model-based approach. *IEEE transactions on consumer electronics*, 66(2):134–143, 2020.

[10] Muhammad Junaid Aamir and Arshad Mansoor. Testing web application from usability perspective. In *2013 3rd IEEE International Conference on Computer, Control and Communication (IC4)*, pages 1–7. IEEE, 2013.

[11] Leonard Adelman and Sharon L Riedel. Usability evaluation. In *Handbook for Evaluating Knowledge-Based Systems*, pages 231–267. Springer, 1997.

[12] Ryan West and Katherine Lehman. Automated summative usability studies: an empirical evaluation. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 631–639, 2006.

[13] Jackson Feijó Filho, Thiago Valle, and Wilson Prata. Automated usability tests for mobile devices through live emotions logging. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 636–643, 2015.

[14] María Paula González, Jesús Lorés, and Antoni Granollers. Enhancing usability testing through datamining techniques: A novel approach to detecting usability problem patterns for a context of use. *Information and software technology*, 50(6):547–568, 2008.

[15] Alaa Mustafa El-Halees. Software usability evaluation using opinion mining. *JSW*, 9(2):343–349, 2014.

[16] Jongwook Jeong, NeungHoe Kim, and Hoh Peter In. Gui information-based interaction logging and visualization for asynchronous usability testing. *Expert Systems with Applications*, page 113289, 2020.

[17] Pavol Fabo and Roman Durikovic. Automated usability measurement of arbitrary desktop application with eyetracking. In *2012 16th International Conference on Information Visualisation*, pages 625–629. IEEE, 2012.

[18] Xavier Ferre, Elena Villalba, Héctor Julio, and Hongming Zhu. Extending mobile app analytics for usability test logging. In *IFIP Conference on Human-Computer Interaction*, pages 114–131. Springer, 2017.

[19] Sandrine Balbo. Software tools for evaluating the usability of user interfaces. In *Advances in Human Factors/Ergonomics*, volume 20, pages 337–342. Elsevier, 1995.

[20] Julián Grigera, Alejandra Garrido, José Matías Rivero, and Gustavo Rossi. Automatic detection of usability smells in web applications. *International Journal of Human-Computer Studies*, 97:129–148, 2017.

[21] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[22] Denys Katerenchuk and Andrew Rosenberg. Rankdcg: Rank-ordering evaluation measure. *arXiv preprint arXiv:1803.00719*, 2018.

[23] William L Hamilton, Jure Leskovec, and Dan Jurafsky. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 2116. NIH Public Access, 2016.

[24] Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.

[25] Carl W Turner, James R Lewis, and Jakob Nielsen. Determining usability test sample size. *International encyclopedia of ergonomics and human factors*, 3(2):3084–3088, 2006.

[26] Daniel W Turner III. Qualitative interview design: A practical guide for novice investigators. *The qualitative report*, 15(3):754, 2010.