

# CohSort

Can automatic cohesion measures improve the readability of summaries by reordering their sentences?

---

**Alfred Sjöqvist, Daniel Tufvesson, Isak Wanström, Karin Stendahl,  
Linus Tullstedt, Ludvig Rammus & Sofia Davidsson**

Supervisors: Robert Eklund & Björn Johansson  
Examinator: Arne Jönsson



## **Abstract**

Extractive text summarization models often generate summaries with fragmented and incohesive sentences. This paper investigates the application of various methods of reordering the sentences in the generated summaries to improve their readability. We developed an application, CohSort, that reorders sentences to maximize the cohesion between sentences in the text according to an aggregate of the LSA-indexes and the L2 readability index in Coh-Metrix. By conducting self-report surveys asking participants to compare original summaries with reordered ones, we found that these reordered summaries were less readable than the originally ordered. This suggests that simply maximizing sentence-to-sentence cohesion for a text does not make it more readable. Further, it suggests that the used cohesion measures do not fully capture readability as a phenomenon.

## **Acknowledgements**

We would like to thank a number of individuals whose involvement with our project has been of great importance. Foremost, we are thankful to our supervisors, Björn Johansson and Robert Eklund, for their guidance and insightful advice, drawn from years of experience in academia. We are also grateful to Evelina Rennes and Daniel Holmer of TextAD for giving us the opportunity to work with this project, and for providing the tools and help necessary to do so. Thanks to Arne Jönsson, for helping us get in touch with the right people and giving advice about technical matters. We would also like to thank Mikael Rosell for guiding us through the information search process and providing us with useful tips. Another thanks to Dżamila Bienkowska, Ehab Abu Sa'a and Martin Andreasson for helping us manage the project and each other. Thanks to Harald Wiltche for helping us think about the philosophy of science and asking questions about the act of asking questions. Finally, a special thank you to all the participants of our study, for patiently answering our surveys and providing us with the data needed.

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Objectives.....	2
1.2 Research question.....	2
1.3 Report structure.....	3
<b>2. Theoretical background.....</b>	<b>4</b>
2.1 Reading comprehension.....	4
2.1.1 Decoding.....	4
2.1.2 Linguistic comprehension.....	4
2.2 Textual features influencing reading comprehension.....	5
2.2.1 Cohesion.....	5
2.2.2 Readability.....	6
2.3 Extractive text summarization.....	6
2.4 TextAD and ElsaSum.....	7
2.5 Text summarization evaluation metrics.....	7
2.5.1 Latent Semantic Analysis.....	8
2.5.1.1 LSA Similarity.....	8
2.5.1.2 LSA Givenness.....	9
2.5.2 L2 Reading Index.....	10
2.5.2.1 Content word overlap.....	11
2.5.2.2 Syntactic similarity.....	11
2.5.2.3 CELEX word frequency.....	12
2.6 BERT.....	12
2.6.1 SBERT.....	13
2.7 Simulated annealing.....	13
2.8 Gram-Schmidt process.....	14
2.9 Syntactic parsing.....	15
2.9.1 Constituency parsing.....	15
2.9.2 Dependency parsing.....	16
2.10 SAPIS.....	17
<b>3. CohSort.....</b>	<b>18</b>
3.1 The CohSort application.....	18
3.2 Parsing written text.....	19
3.3 SBERT for sentence embeddings.....	19
3.4 Implementing LSA Similarity.....	19
3.5 Implementing LSA Givenness.....	20
3.5.1 Simplified Gram-Schmidt for constructing the hyperplane.....	20
3.5.2 Projecting onto the hyperplane and computing the indexes.....	20
3.6 Implementing L2 Reading Index.....	21
3.6.1 Content word overlap.....	21

3.6.2 Syntactic similarity.....	21
3.6.2.1 Syntactic similarity with dependency relations.....	21
3.6.2.2 Syntactic similarity with constituency trees.....	22
3.6.2.3 Analyzing the trees.....	22
3.6.3 CELEX word frequency.....	23
3.7 Putting the indexes together.....	23
3.8 Reordering sentences.....	24
<b>4. Research design and procedure.....</b>	<b>25</b>
4.1 Design.....	25
4.1.1 Materials.....	25
4.1.1.1 Articles to be summarized.....	25
4.1.1.2 ElsaSum and CohSort.....	25
4.1.1.3 Survey.....	26
4.1.1.4 SAPIS.....	27
4.1.2 Participants.....	27
4.1.3 Ethical considerations.....	27
4.2 Procedure.....	27
4.2.1 Analysis.....	28
4.2.2 Technical evaluation.....	29
<b>5. Results.....</b>	<b>32</b>
5.1 Survey article 1 summary.....	32
5.2 Survey article 2 summary.....	35
5.3 Survey article 3 summary.....	38
5.4 Mean readability for each article.....	41
5.5 Technical evaluation.....	44
<b>6. Discussion.....</b>	<b>46</b>
6.1 Survey discussion.....	47
6.2 Technical evaluation.....	49
6.3 Future work.....	50
6.4 Implications.....	51
6.5 Ethics and sustainability in relation to text summarization.....	51
<b>7. Conclusions.....</b>	<b>53</b>
<b>Bibliography.....</b>	<b>54</b>
<b>Appendix A.....</b>	<b>57</b>
<b>Appendix B.....</b>	<b>58</b>
<b>Appendix C.....</b>	<b>62</b>
<b>Appendix D.....</b>	<b>67</b>
<b>Appendix E.....</b>	<b>70</b>



# 1. Introduction

The ability to access and understand written information is crucial in today's society, where an increasing amount of information is being made available for larger portions of the population. As such follows that, for those individuals with reading difficulties, such as dyslexia, information may not be as easily accessible as for those without these hindrances. The ability to process and comprehend text varies between individuals, and highlights the need for innovative solutions that can help improve the readability and accessibility of written information, so that it better suits each person's needs. This can help enhance their learning experience, and ultimately their ability to participate in the modern information-driven society.

Language technology, a subfield of artificial intelligence dedicated to the task of processing natural language using computational methods, sees great potential when it comes to addressing the challenge of making information accessible to every reader. A key area of language technology is natural language processing (NLP), which focuses on enabling computer programs to analyze, understand, and generate natural language. This task requires understanding of the syntactic, semantic, and pragmatic aspects of language. Syntactic analysis involves identifying the grammatical structure of sentences, while semantic analysis focuses on extracting the meaning of words, phrases, and sentences. Pragmatic analysis considers the context of which language is used to figure out meanings and intentions.

Applications of NLP have shown promising results when it comes to meeting the need for improved text accessibility (Margarido et al., 2008). One such application is extractive text summarization, which has the potential to simplify text according to various metrics, in order to make them more accessible and easier to comprehend for individuals with reading difficulties. The Text Adaptation for Increased Reading Comprehension (TextAD) research project at Linköping University aims to develop such a tool, one iteration being the ElsaSum summarizer (see Section 2.4).

There are numerous definitions of readability, and in the context of this project, we refer to readability as the degree of ease with which a reader can comprehend written text. This definition is perhaps closest to that of Dale and Chall (1949, p. 23):

The sum total (including all the interactions) of all those elements within a given piece of printed material that affect the success a group of readers have with it. The success is the extent to which they understand it, read it at optimal speed, and find it interesting. (Dale and Chall, 1949, p. 23)

Readability, in this sense, is therefore influenced by factors such as language complexity, text structure, and the reader's prior knowledge of the subject matter. One relevant factor which can influence readability, as explained by Graesser et al. (2004), is cohesion. This refers to the explicit linguistic features of a text (e.g., words, phrases and sentences) that subsequently cue the reader into a formation of coherent mental representations of the content. This differs

from coherence, which is a distinct term in that it represents the psychological construct of these mental representations, which are also dependent on the reader's skills and prior knowledge. Cohesion is an objective property of the text itself, making it relevant in the context of computational approaches to text adaptation. Its objective nature allows for algorithmic analysis and manipulation of text in a consistent and accurate manner.

In this project we aim to improve the readability of summaries generated by ElsaSum, through modifications that serve to maximize their cohesion. To do this, the Coh-Metrix framework will be utilized, which is a tool for analysis of text characteristics such as text cohesion and readability (Graesser et al., 2004). A new application, CohSort, will be developed from the foundation of the ElsaSum summarizer, and evaluated through a comparative study design utilizing a survey for data collection. Our hope is that the results will provide further insight into the role of cohesion measures within the topic of extractive text summarization, and ideally a more effective solution for TextAD that contributes to the usefulness of the tool in readability assistance for those who need it.

## **1.1 Objectives**

The main objectives of this project are as follows:

- To implement modified Coh-Metrix LSA indexes that rely on SBERT sentence embeddings as opposed to Singular Value Decomposition.
- To implement a Coh-Metrix L2 Reading Index for Swedish texts.
- To create an application that sorts sentences of summaries as to maximize their cohesion given by an aggregate of the implemented LSA indexes and the L2 Reading Index.
- To evaluate if summaries sorted in this way actually improves readability.

## **1.2 Research question**

In order to achieve the objectives of this project, as well as to direct our research, the following research question is proposed:

Can automatic cohesion measures improve the readability of summaries by reordering their sentences?

By addressing this research question, we aim to explore the effectiveness of the modified LSA measure and the L2 Reading Index in increasing the readability and cohesion of summaries generated by extractive text summarization. This will provide valuable insights into how these cohesion measures can contribute to the overall goal of improving text accessibility for individuals with reading difficulties.

## 1.3 Report structure

This report is structured into seven chapters. Chapter 1 introduces the project with a brief background of its purpose, as well as its objectives and research question. Chapter 2 covers the theoretical framework for the project, exploring aspects of reading comprehension, text summarization, and evaluation metrics. This chapter also lays the groundwork for subsequent chapters by overviewing technical concepts such as BERT, SBERT, simulated annealing, the Gram-Schmidt process, and syntactic parsing. Following this, Chapter 3 delves into the application developed for the project, CohSort, integrating the theory discussed in Chapter 2. The application is described in its entirety, along with the process of implementing various features, concluding with testing of the application. Chapter 4 outlines the methodology of the study, covering the design aspects, materials used, participants involved, as well as ethical considerations. The procedure is detailed, including the analysis and statistical tests conducted. Chapter 5 presents the results obtained from the study, before being discussed comprehensively in Chapter 6. Lastly, Chapter 7 concludes the project report by summarizing the key findings.

## 2. Theoretical background

This chapter covers the theoretical framework for the project, exploring aspects of reading comprehension, text summarization, and evaluation metrics. It also overviews technical concepts such as BERT, SBERT, simulated annealing, the Gram-Schmidt process, and syntactic parsing.

### 2.1 Reading comprehension

Reading comprehension refers to the ability for understanding and interpreting written text, a task which incorporates several cognitive processes. Two fundamental components of reading comprehension are decoding and linguistic comprehension. Decoding refers to the ability of transforming written symbols into their corresponding sounds, while linguistic comprehension is the ability to interpret and understand spoken language. These two abilities can be mathematically represented as in Equation 2.1 (Gough and Tunmer, 1986):

$$R = D \times C \quad (2.1)$$

where  $R$  denotes reading comprehension,  $D$  denotes decoding, and  $C$  represents linguistic comprehension. Each variable has a range from 0 to 1, going from non-existent skill to perfect skill, leading to the assertion that either of these abilities being absent results in limited reading comprehension, regardless of skill-level in the other component.

#### 2.1.1 Decoding

The ability to decode relies on having a fundamental understanding of the correlation between symbols and sounds in language, something Gough and Tunmer (1986) call the orthographic cipher. They argue that knowledge of this cipher is crucial in an alphabetic orthography system such as English. A reader who is a capable decoder can quickly recognize words without relying on context, and can also accurately read novel words, such as pseudowords (words that can be pronounced but do not have any inherent meaning), by applying the rules of the orthographic cipher. While the system is not infallible, as a language such as English is not perfectly phonetic, containing various irregular words that do not conform to the general rules, understanding of the orthographic cipher remains a crucial prerequisite for proficient reading comprehension. If written text cannot be transformed into spoken language, it cannot be comprehended (Gough and Tunmer, 1986).

#### 2.1.2 Linguistic comprehension

Decoding is an essential skill for effective reading comprehension, although it is not sufficient on its own, as made clear by Equation 2.1. Once words are decoded, they must also be understood in terms of linguistic comprehension. This term differs from reading comprehension, which is the broader concept of understanding what you read, facilitated by the two constituent parts that are decoding and linguistic comprehension. Linguistic

comprehension refers to the ability of interpreting sentences and discourses through given lexical information. It encompasses the understanding of syntaxes, what words and grammatical structures mean, and to be able to perceive the contextual or thematic elements of a dialogue. Essentially, to be able to make sense of written information. The need for this ability in reading can be exemplified through the way in which a person could possess the ability to decode a foreign language without understanding it. Or as with a child that could decode words without being able to comprehend their meaning (Gough and Tunmer, 1986).

## **2.2 Textual features influencing reading comprehension**

Textual features play an important role in shaping reading comprehension, as they are what forms the structure of a text, facilitating an understanding of the content within it through means such as decoding and linguistic comprehension. These features can range from the general organization of the text, to the syntactical arrangement of its words and phrases. Two key textual features that relate to reading comprehension in this sense are cohesion and readability.

### **2.2.1 Cohesion**

As discussed briefly in the introduction (see Chapter 1), with reference to Graesser et al. (2004), cohesion is characterized by the explicit linguistic features within a text that guide the reader towards constructing coherent mental representations of the content. These features include words, phrases, and sentences that interconnect to create an unified, cohesive whole. Here it is important to distinguish between cohesion and coherence, with the latter being a psychological construct of mental representations that is dependent on the reader's skills and prior knowledge of the subject. Cohesion is an objective property inherent to the text itself, which in turn has an effect on coherence in the way that it cues the formation of these mental representations.

Key to text cohesion is the degree of connectivity between adjacent sentences, as explained by Linderholm et al. (2000). This connectivity can be reflected in the level of conceptual overlap between sentences, as well as through the presence of specific cues, such as connectives, that help readers in linking together ideas across sentences. Furthermore, cohesion can also be influenced by the overall organization of a text, expressed through the temporal and causal sequence of events.

Cohesive features collectively contribute to upholding coherence in the text by reducing the need for prior topic-specific knowledge, through accentuating self-containment within the text. With maintaining text coherence through these features of cohesion comes the subsequent facilitation of reading comprehension. McNamara et al. (1996) explains that increase of coherence in a text also entails that the reader will comprehend it better. They conducted research that specifically indicates that text coherence can be most beneficial for readers with less prior knowledge. While less coherent texts demand more prior knowledge of the topic, highly coherent texts are more self-contained, thus requiring less topic-specific knowledge making them more comprehensible for the reader.

## 2.2.2 Readability

Cohesion has an influence on text readability, another aspect related to reading comprehension. Readability, as briefly discussed in the introduction (see Chapter 1), is a measure of the ease with which a reader can understand a written text. As explained by Crossley, Greenfield and McNamara (2008), this has traditionally been based on surface-level linguistic features that often fail to account for the cognitive processes employed by reader's interacting with a text. Measures such as the Flesch-Kincaid grade level and Flesch reading ease have predominantly focused on mechanical aspects of text, such as sentence length, words per sentence, word frequency, and syllables per word. This does not factor in the deeper levels of text processing, such as cohesion.

A more nuanced approach to assessing readability comes through the Coh-Metrix framework, referenced briefly in the introduction (see Chapter 1). Coh-Metrix is a computational tool for measuring text cohesion and difficulty, providing a detailed analysis of not just surface-level linguistic features, but also cognitive reading processes such as understanding, decoding, and syntactic parsing. Crossley, Greenfield and McNamara (2008) found that predicting readability was more accurate with three particular Coh-Metrix variables: lexical coreferentiality, syntactic sentence similarity, and word frequency. These variables, which are related to cognitive reading processes and psycholinguistic theory, outperformed the more traditional measures in their predictive capacity. This indicates the value of incorporating Coh-Metrix measures when evaluating text readability, to provide a more comprehensive assessment (Crossley, Greenfield and McNamara, 2008).

## 2.3 Extractive text summarization

One application of NLP is automatic text summarization (ATS), which is the task of automatically producing summaries of a source document. The goal of ATS is to shorten a document, while preserving the most relevant information. There are plenty of ways to do this, and text summarization can be classified into different categories (Orăsan, 2019). One of these categories is whether a summary is abstractive or extractive. An abstractive summary has new content not found in the source document, while an extractive summary only contains content from the source document. The extractive summary is produced by extracting the most relevant sentences of the source document and concatenating them to obtain a shorter text. The sentences themselves are not manipulated in any way and remain identical to the sentences in the source document (Hahn and Mani, 2000). Since this approach does not take the cohesion of the text into account, it can result in the summaries being difficult to read. The lack of cohesion in the summary sentences can be a result of not taking anaphora, repetitions in the beginnings of sentences, into account (El-Kassas et. al. 2021). Furthermore, some information might not be in the form of text, but rather in tables or illustrations, posing a significant challenge for extractive summarization tools (Hahn and Mani, 2000).

In general, ATS systems can be divided into three tasks: pre-processing, processing and post-processing. In the pre-processing task, a representation of the source document is created. This can involve techniques such as tokenization, stop word removal and stemming. During processing, a summarization technique is used to convert the source document to a summary. In an extractive text summarizer, the processing task consists of three parts. Firstly, a representation of the text is created. Secondly, the representation is used to score the sentences. Finally, the highest scoring sentences are extracted and concatenated, resulting in a summary. Post-processing involves correcting issues in the generated summary, such as reordering the extracted sentences (El-Kassas et. al. 2021).

A study by Monsen and Rennes found that extractive summaries were perceived to be more fluent and adequate than abstractive summaries (Monsen and Rennes, 2022). While abstractive summaries often contained factual errors, extractive summaries were found to adhere more closely to the original text, preserving its factual content.

## **2.4 TextAD and ElsaSum**

The TextAD research project at Linköping University aims to improve the understanding of different reading difficulties and use this knowledge to develop tools for automatic adaptation of texts to suit the individual reader's needs. People with reading difficulties are a diverse group and the issues they face as well as the degree of difficulty vary from person to person. Hence, there is no such thing as an easy to read text that suits every reader (Isaksson, 2021).

ElsaSum, an extractive text summarizer, was developed within the TextAD project by Andersson (2022). The model was trained using a dataset composed of 349,935 news articles from the Swedish newspaper, Dagens Nyheter. This dataset was initially created by Monsen and Jönsson (2021), and subsequently refined by Monsen and Rennes (2022).

The summarizer takes raw text as input, ranks the sentences according to the model's understanding of the most important or informative parts of the text, and outputs a summary consisting of the highest ranked sentences. The order of these sentences are the same in the summary as in the source document. Development efforts have been directed towards a post-processing component within ElsaSum; however, this element has not yet been subjected to evaluation or testing. The output of the summarizer is what we intend to reorder for the purpose of this project. It is important to note that there will be no code integration between the two; they will maintain operational independence.

## **2.5 Text summarization evaluation metrics**

There are two types of methods to evaluate automatically generated summarizations. Intrinsic evaluations use human evaluation and focus on the summary's cohesion and informativeness. Extrinsic evaluation methods use some task-based performance measure to evaluate the quality of a summary (El-Kassas et. al. 2021). One automatic tool for such evaluations is

Coh-Metrix, as described in Chapter 1 and Section 2.2.2. It consists of more than 200 measures of readability, cohesion and language (Graesser et al., 2004). For this project we have implemented measures concerning Latent Semantic Analysis and readability.

## 2.5.1 Latent Semantic Analysis

Latent Semantic Analysis (LSA) consists of analyzing the semantics of written text using word embeddings (Landauer, 2013). A word embedding is a vector representation of a word. These embeddings are computed based on their statistical co-occurrence with other words in the training data, thereby capturing the contexts in which they often occur. This makes the vector space of the embeddings semantically meaningful, that is, embeddings that are located close to each other are often semantically similar. For example, the embeddings for “cat” and “dog” may be located close to each other since these words frequently co-occur. Embeddings are not limited only to words, but may also be computed for entire sentences and even paragraphs. For example, the sentences “I want an ice cream” and “I want a popsicle” both have embeddings that are located close to each other.

LSA typically uses Singular Value Decomposition (SVD), a statistical method, for computing embeddings (Landauer, 2013). This allows the dimensionality of the embeddings to be reduced. Rather than using embeddings with dimensions the size of the entire trained vocabulary, with one element for each unique word, the dimensions can be reduced to typically 200–500 elements. As a consequence, the embeddings end up containing estimates of the similarities between every word; even words that do not co-occur in the data. Non-reduced embeddings, in contrast, contain only the estimates of similarity between words that co-occur. From this, LSA can induce the meaning of words that do not co-occur in the data, which means it can, for example, compare the similarity between two texts that contain completely different words.

### 2.5.1.1 LSA Similarity

Computation of the similarity between two embeddings (either word, sentence or paragraph embeddings) is done with the cosine between the two embeddings:

$$\text{Similarity}(S_v, S_u) = \cos[v, u] = \frac{v \cdot u}{|v| |u|} \quad (2.2)$$

where  $v$  and  $u$  are the two embeddings,  $v \cdot u$  is the dot product between them, and  $|v|$  and  $|u|$  are their respective Euclidean lengths. The computed value ranges from -1 to 1. 1 indicating complete similarity (synonymy), -1 complete opposites (antonymy), and 0 no semantic relation. For example, “car” and “automobile” would have a cosine close to 1 since they are more or less synonymous. “Car” and “galaxy” are not related and would have a cosine close to 0. “Big” and “small” are antonyms and would therefore have a cosine approaching -1.

At the time of writing, Coh-metrix provides eight LSA indices (University of Memphis, n.d.), six of which were implemented in this project.

LSASS1 measures the mean similarity between adjacent sentences:

$$LSASS1 = \frac{1}{n-1} \sum_{i=1}^{n-1} \text{Similarity}(S_i, S_{i+1}) \quad (2.3)$$

LSASS1d is like LSASS1 but measures the standard deviation rather than the mean:

$$LSASS1d = \sqrt{\frac{\sum_{i=1}^{n-1} (\text{Similarity}(S_i, S_{i+1}) - LSASS1)^2}{n-1}} \quad (2.4)$$

LSASSp measures the mean similarity between all possible sentence pairs in the text:

$$LSASSp = \frac{1}{n^2-1} \sum_{i=1}^{n-1} \sum_{k=1}^{n-1} \text{Similarity}(S_i, S_k), i \neq k \quad (2.5)$$

LSASSpd is like LSASSp but measures the standard deviation:

$$LSASSpd = \sqrt{\frac{\sum_{i=1}^{n-1} \sum_{k=1}^{n-1} (\text{Similarity}(S_i, S_k) - LSASSp)^2}{n^2-1}}, i \neq k \quad (2.6)$$

### 2.5.1.2 LSA Givenness

LSAGN and LSAGNd are givenness measures, that is, they measure how much new semantic information a sentence provides in relation to the previous sentences (McNamara et al., 2014, p. 66–67). This is done by first constructing a hyperplane  $\mathbb{V}$  from the embeddings of the previous sentences:

$$\mathbb{V} = [v_1, v_2, \dots, v_{k-1}] \quad (2.7)$$

The hyperplane represents all the prior information. The embedding  $v_k$  of the current sentence is then projected onto the hyperplane, resulting in a projected vector  $v_{given}$  which represents the given information contained in the sentence.

$$v_{given} = \text{Project}_H(v_k, \mathbb{V}) \quad (2.8)$$

Projecting onto a hyperplane is done by projecting the  $v_k$  onto each vector in the hyperplane, and then summing the projections.

$$\text{Project}_H(v_k, \mathbb{V}) = \sum_{v \in \mathbb{V}} \text{Project}_V(v_k, v) \quad (2.9)$$

where  $project_v(v_k, v)$  is the projection of  $v_k$  onto  $v$ :

$$Project_v(v_k, v) = \frac{v_k \cdot v}{|v|^2} v \quad (2.10)$$

To compute the new information, the hyperplane projection is subtracted from the current embedding:

$$v_{new} = v_k - v_{given} \quad (2.11)$$

The givenness is the proportion between the new and given information and could therefore be calculated as:

$$givenness = \frac{|v_{given}|}{|v_{new}| + |v_{given}|} \quad (2.12)$$

where  $|v_{given}|$  and  $|v_{new}|$  are the Euclidean lengths of the vectors. A *givenness* value approaching 1 indicates high givenness, that is, most information was already given by the previous sentences and the current sentence contains less new information. A low value approaching 0 indicates plenty of new information in the current sentence. New information leads to lower cohesion, while familiar information leads to higher.

The LSAGN index consists of computing the mean givenness for each sentence except the first in the text. Givenness can not be computed for the first sentence, since givenness requires there to be previous sentences to compare with.

$$LSAGN = \frac{1}{n-1} \sum_{i=2}^n givenness_i \quad (2.13)$$

The LSAGNd index is like the LSAGN index but computes the standard deviation:

$$LSAGNd = \sqrt{\frac{\sum_{i=2}^n (givenness_i - LSAGN)^2}{n-1}} \quad (2.14)$$

## 2.5.2 L2 Reading Index

The Coh-Metrix L2 Reading Index is a readability formula designed to reflect the cognitive and psycholinguistic processes involved in reading, particularly for second language readers (Crossley et al., 2008; McNamara et al., 2014, p. 80–81). The measure takes into account the content of a given text, as well as its complexity, and is calculated according to the following formula:

$$\begin{aligned} L2 = & -45.032 + 52.230 \times ContentWordOverlap \\ & + 61.306 \times SentenceSyntaxSimilarity \\ & + 22.205 \times CELEXWordFrequency \end{aligned} \quad (2.15)$$

The formula incorporates three other Coh-Metrix indexes which are calculated together with constants. Content Word Overlap represents the unweighted proportion of content words shared by adjacent sentences. Sentence Syntax Similarity measures the mean syntactic similarity between adjacent sentences. CELEX Word Frequency refers to the mean minimum logarithm frequency for content words for each sentence.

### 2.5.2.1 Content word overlap

The content word overlap index (CRFCWO1) measures how often content words overlap between adjacent sentences (McNamara et al., 2014, p. 65). The following formula can be used for computing this:

$$MeanContentOverlap = \frac{1}{n-1} \sum_{i=1}^{n-1} ContentWordOverlap(S_i, S_{i+1}), \quad (2.16)$$

where  $n$  is the number of sentences in the text.

The content word overlap between two adjacent sentences is computed as a proportion between the overlapping content words and the sum total of words in both sentences:

$$ContentWordOverlap(S_1, S_2) = \frac{\#OverlappingWords(S_1, S_2)}{\#Words(S_1) + \#Words(S_2)}, \quad (2.17)$$

where  $\#OverlappingWords(S_1, S_2)$  is the number of overlapping content words in the two adjacent sentences  $S_1, S_2$ , and  $\#Words(S_i)$  is the number of words in a particular sentence.

### 2.5.2.2 Syntactic similarity

The Sentence Syntax Similarity index, or more specifically the SYNSTRUTa index, measures the mean syntactic similarity between adjacent sentences (McNamara et al., 2014, p. 71–72). This can be computed with the following formula:

$$MeanSimilarity = \frac{1}{n-1} \sum_{i=1}^{n-1} AdjacentSimilarity(S_i, S_{i+1}) \quad (2.18)$$

where  $n$  is the number of sentences in the text.

The syntactic similarity between two sentences is computed by comparing the nodes in their constituency trees. This can be computed with the following formula:

$$AdjacentSimilarity(S_1, S_2) = \frac{\#NodesCommon(S_1, S_2)}{\#Nodes(S_1) + \#Nodes(S_2) - \#NodesCommon(S_1, S_2)}, \quad (2.19)$$

where  $\#NodesCommon(S_1, S_2)$  is the number of nodes in the largest common constituency tree of the two adjacent sentences  $S_1$  and  $S_2$ , and  $\#Nodes(S_i)$  is the number of nodes in the constituency tree of sentence  $S_i$ . See Figure 2.1 for an example.

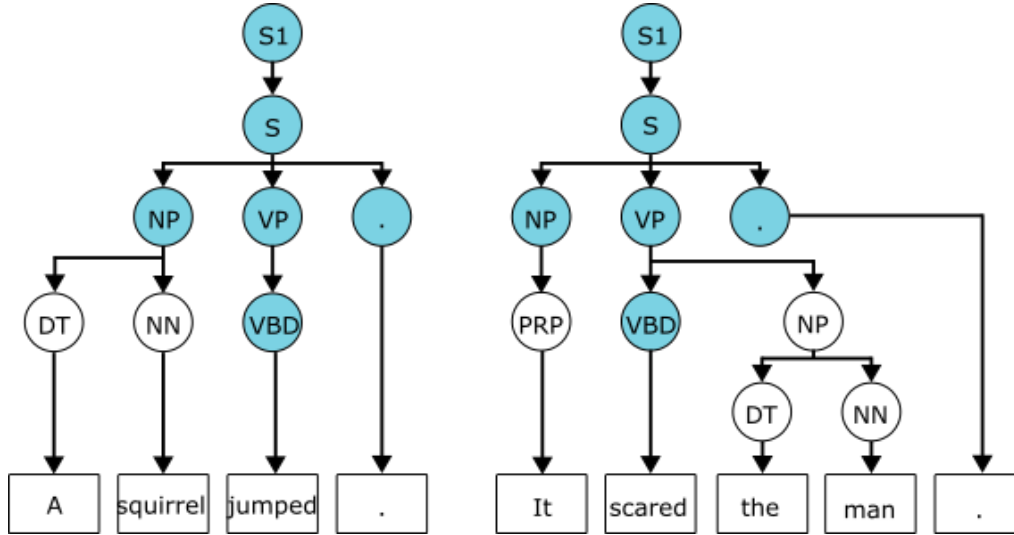


Figure 2.1: Two sentences with their respective constituency trees. The blue nodes highlight their common tree. Here the adjacent similarity between the two sentences is  $4 / (8 + 10) = 0.22\dots$  Example is based on McNamara et al. (2014, p. 71–72).

### 2.5.2.3 CELEX word frequency

The CELEX word frequency index measures the occurrence of words in the English language (McNamara et al., 2014, p. 73). More common words are easier to cognitively process than uncommon. The frequencies are word counts taken from the CELEX database, which is a corpus of English texts. There are several word frequency indexes, but the one used in this project is the WRDFRQmc index, which selects the least frequent words in each sentence and computes the average of these.

$$CELEX\ Word\ Frequency = \frac{1}{n} \sum_{k=1}^n \log(LeastFrequent(S_i)), \quad (2.20)$$

where  $LeastFrequent(S_i)$  is the frequency of the least rarest word in the sentence  $S_i$ . Note that the frequencies are log-transformed. This is because lower frequencies matter more than higher frequencies for readability.

## 2.6 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained language model which can be applied to many different NLP tasks (Devlin et. al., 2019). It requires

only fine-tuning for the specific task at hand. This is in contrast to the traditional approach where the model must be trained from scratch for the specific task.

The power of BERT comes from its bidirectional pre-training, that is, during pre-training it is provided the full context in which each word appears in (Devlin et. al., 2019). Previous approaches consisted of unidirectional pre-training where the model is only provided either left-hand context or the right-hand context; either the words to the left or to the right of the target word. Bidirectional pre-training, however, provides both the left and right words. To pre-train the model in this bidirectional sense, Devlin et. al applied a masked language model (MLM) approach, where a hidden word in a sentence is to be guessed given the other words.

While the MLM training gives BERT an understanding of how words relate to each other, it is not sufficient for how sentences relate to each other. This is something that is necessary for many NLP tasks. BERT was therefore, in addition to MLM training, trained on a “next sentence prediction” task (Devlin et. al., 2019). Here a pair of sentences is presented to BERT. 50% of the time the pair consists of sentences that follow each other, and 50% of the time the following sentence is picked randomly. BERT is then tasked to predict whether the following sentence is the actual next sentence or not. In this way BERT learns the relationships between sentences. That is, which sentences are related to each other and which are not.

## **2.6.1 SBERT**

One of the many NLP tasks which BERT can be used for is computing word embeddings, which refers to the conversion of words into a vector space where level of semantic similarity dictates the proximity of words to each other. While BERT is suitable for computing such word embeddings, it falls short when it comes to computing sentence embeddings, that is vector representations of full sentences as opposed to singular words. BERT cannot compute semantically meaningful sentence embeddings. Sentence-BERT (SBERT), a modification of BERT, addresses this limitation by employing siamese and triplet network structures, making it capable of producing semantically meaningful sentence embeddings, that are furthermore conveniently comparable through cosine similarity. This modification drastically reduces the computational effort and time needed for the task of locating most similar pairs from a large set, while maintaining the accuracy of BERT. Thus, SBERT is the preferable choice for tasks involving comparison of large-scale semantics similarity, clustering, and information retrieval (Reimers and Gurevych, 2019).

## **2.7 Simulated annealing**

Search problems can be defined as a set of search states and a set of actions for moving from one state to another. A state is thereby connected to a number of neighboring states. These states and connections make up the problem’s search space. Search algorithms are employed to find the solution state by navigating the search space in an efficient manner. The choice of search algorithm depends on the search problem.

For approximating a solution state in a large search space, one may employ simulated annealing. This search algorithm combines the benefits of random walk with that of hill climbing (Russell & Norvig, 2016, p. 125). Hill climbing, which is to continually move to the best neighboring state, is an efficient straightforward way of finding the solution. However, it does not guarantee finding the solution, since it may get stuck in local maxima; these are search states that have no better neighboring states, yet they are not the solution. Random walk, on the other hand, never gets stuck in local maxima, but is extremely inefficient at finding the solution. Simulated annealing combines hill climbing with random walk, by essentially starting out with a random walk and then slowly transitioning into a hill climbing search. This process is specified in Figure 2.2.

```
function simulated_annealing(problem, schedule)
    inputs: problem, the search problem.
           schedule, a table of the temperature over time.
    output: a solution state.

    current_state = problem.initial_state
    for time = 1 to  $\infty$ 
        temperature = schedule(time)
        if temperature = 0
            return current_state
        next_state = random_successor_state(current_state)
         $\Delta E$  = score(next_state) - score(current_state)
        if  $\Delta E > 0$ 
            current_state = next_state
        else
            current_state = next_state with probability  $e^{\Delta E / \text{temperature}}$ 
```

Figure 2.2: The simulated annealing algorithm. As the “temperature” decreases over time, the randomness decreases. Pseudocode is based on Russell & Norvig (2016, p. 126).

## 2.8 Gram-Schmidt process

The Gram-Schmidt process is a method for constructing an orthonormal basis (ON-basis) from a set of linearly independent vectors (Janfalk, 2019, p. 150-156). A basis is a set of vectors that can be combined to describe any vector in a vector space. This set of vectors is said to be the basis of the space. An ON-basis is a basis, where each base vector is orthogonal to each other and is of length 1. That a vector is orthogonal to another vector means that their cosines are zero. The vectors of the ON-basis are said to be orthonormal.

The Gram-Schmidt process is an iterative process, where each vector is orthonormalized (that is, turned into an orthonormal vector) and added to the new ON-basis. After the process is complete, the resulting ON-basis defines the same space as the original set of vectors, but with orthonormalized vectors.

The process can be described with the following steps:

1. Select the first vector from the set of vectors.
2. Normalize the vector.
3. Fill out the ON-base by adding the normalized vector to it.
4. Select the next vector.
5. Project the vector onto the vectors specifying the so far constructed ON-base. This can be done using Equation 2.9, where  $\mathbb{V}$  is the ON-base and  $v_k$  is the vector.
6. Subtract the projection from the vector.

The steps from including 2 to including 6 are repeated until all the original vectors have been processed. The resulting vectors are all orthonormalized in relation to each other and specify the ON-basis for the space of the original vectors.

## 2.9 Syntactic parsing

Syntactic parsing is the process of analyzing the structure of a sentence in order to identify its components and how they relate to each other. This is accomplished by constructing a representation of the syntactic structure of a sentence. The syntactic structure can be represented as constituencies or dependencies, which both can be represented as trees (Jurafsky and Martin, 2023).

### 2.9.1 Constituency parsing

Constituencies are the grammatical units, typically phrases, that make up sentences. These can generally be rearranged within the sentence without altering its meaning. Constituency parsing analyzes a sentence by identifying these phrases and their relationships, resulting in a hierarchical tree. The parse tree starts with the full sentence in the top node, and is divided into sub-trees that represent the constituencies of the sentence (Jurafsky and Martin, 2023). Consider for example the sentence “I saw a ball yesterday”. At the first level, it consists of the noun phrase “I” and the verb phrase “saw a ball yesterday”. At the next level, the verb phrase is further broken down into the verb phrase “saw”, the noun phrase “a ball” and the adverbial phrase “yesterday”. Lastly, the phrases are broken down into the parts of speech they consist of. This is illustrated in Figure 2.3.

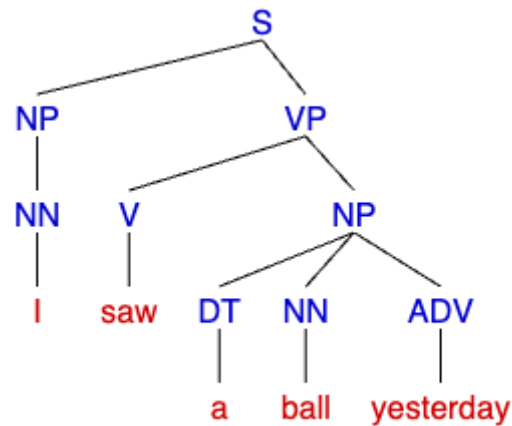


Figure 2.3: A parse tree for the sentence “I saw a ball yesterday”.

## 2.9.2 Dependency parsing

A dependency is a directed relationship between two words: a head and a dependent (Jurafsky and Martin, 2023). Typically, the head word determines syntactic or semantic attributes of the dependent word. Each word has exactly one head, except for the root word of the sentence which has none. In general, the root word is the main verb of the sentence around which the rest of the words are structured. With dependency parsing, the dependency relations of a sentence are analyzed. Each dependency relation has a dependency label, which describes the type of relation the dependent has to its head word. The relations of a sentence can be represented as a dependency tree, where each word is represented as a node and each dependency is represented by a directed arrow. For example, the dependencies in the sentence “I saw a ball yesterday” are as follows:

- “saw” is the root word
- “I” and “ball” are dependents of “saw”
- “the” modifies “ball”
- “ball” is the direct object of “saw”
- “yesterday” modifies “saw”

The dependency tree is illustrated in Figure 2.4.

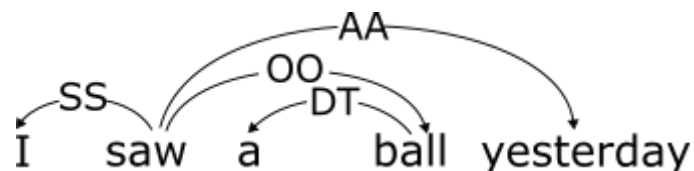


Figure 2.4: An example of a dependency tree for a sentence.

## 2.10 SAPIS

SAPIS (Simplification and Analysis of Texts API Service) is an API service that offers tools and techniques to simplify and analyze textual data, with the goal of improving text accessibility. It is designed to be accessed from a remote server, allowing easy implementation of new services. SAPIS targets both professional users, such as web editors, as well as everyday users interested in text simplification and analysis (Fahlborg and Rennes, 2016).

Currently, SAPIS provides four running services: STILL-ETT, LEXICAL METRICS, SURFACE METRICS, and STRUCTURAL METRICS. Users can specify which of these services to run and provide corresponding arguments through an input JSON object, the resulting metrics and simplification suggestions are merged and returned to the client as a JSON object (Fahlborg and Rennes, 2016).

STILL-ETT is a rule-based text simplification tool for Swedish, it provides services such as rewriting to passive-to-active, quotation inversion, rearranging to straight word order, sentence splitting, and synonym replacement. SCREAM (Swedish Compound READability Metric) provides readability statistics for Swedish texts, including surface metrics like LIX, OVIX, nominal ratio, average sentence length, and average word length. LEXICAL METRICS offers a categorized frequency analysis of word occurrences using the SweVoc dictionary, indicating commonly used words in an easy-to-read text. STRUCTURAL METRICS provides syntactic and morpho-syntactic features based on part-of-speech tags and dependency tags (Fahlborg and Rennes, 2016).

## 3. CohSort

This chapter presents the application CohSort and its implementation. The purpose of this application is to reorder sentences of generated summaries, in order to maximize the sentence-to-sentence cohesion. These summaries will then be used for determining if reordering sentences is a viable approach for improving readability. The chapter provides a detailed account of CohSort's structure and constituent parts, the process through which they were implemented, followed by testing and evaluation of the application.

### 3.1 The CohSort application

The application reorders the sentences of a text summary to maximize the cohesion of the summary. It takes a written Swedish text summary as input. Technically, the text does not strictly have to be a summary, as the application makes no distinction between a regular text and a summary. From this input it produces a new text as output, containing the identical sentences as the input text, but with their order modified (see Figure 3.1).

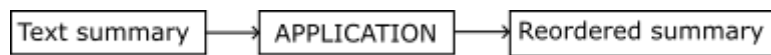


Figure 3.1: The application receives a summary as input, and produces as output a reordered summary with higher cohesion.

The order is determined by finding the order which receives the highest cohesion score. Finding the correct order is handled as an optimization problem, which we solve using simulated annealing (see Section 3.8). The cohesion score is an aggregate of the LSASS1, LSASS1d, LSAGN, LSAGNd, SYNSTRUta, and CRFCWO1 indexes. Computing the aggregate cohesion score is described in Section 3.7.

Figure 3.2 illustrates the execution order of the application, from input to output. The structure of the application and its components are illustrated in Appendix A.

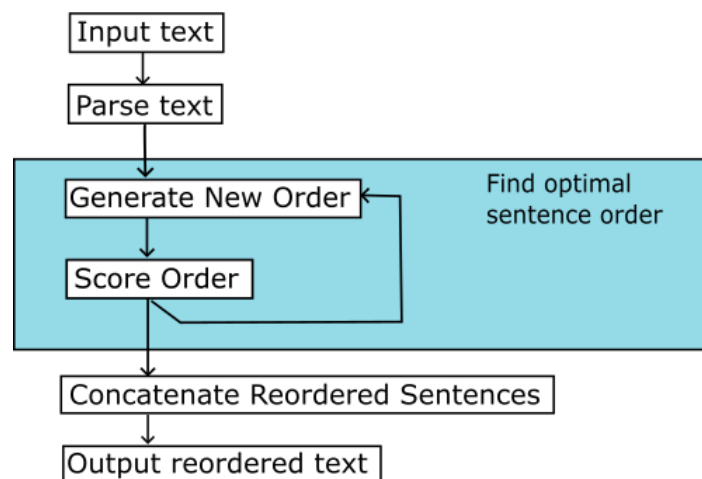


Figure 3.2: The order of execution of the CohSort application.

## 3.2 Parsing written text

To parse the input text we employed Stanza which is a Python package for natural language analysis (Stanford NLP Group, 2020), as well as Berkeley Neural Parser (Benepar) which is a constituency parser for Python (Kitaev, 2018). Stanza was used for sentence segmentation, tokenization, POS-tagging, and dependency parsing. Stanza produces a Document object that contains parsed NLP data in the form of Word, Sentence, and Token data objects. These objects conveniently structure the data, so we therefore used these for handling the NLP data across the application.

Benepar was used for constituency parsing. Stanza has a constituency parser, but at the time of this project, however, it did not support Swedish texts. We therefore used Benepar for this task. While Benepar can be used for parsing raw text, we instead fed it the already parsed words and their respective POS-tags, to make sure the resulting constituency trees were consistent with the Stanza parsed text data. These parsed constituency trees were then attached as a custom property to each Sentence object, allowing for easy access across the application.

## 3.3 SBERT for sentence embeddings

To compute the sentence embeddings for LSA (see Section 2.5.1) we employed SBERT (see Section 2.6.1) instead of the conventional SVD method. We used a Swedish SBERT model *sentence-bert-swedish-cased v2.0* provided by The National Library of Sweden (Rekathati, 2023). This allowed us to compute embeddings for Swedish sentences.

The embeddings are computed before being used in the LSA-indexes. In our application, however, the same embeddings are being used several times in the LSA-indexes. This means that the embeddings must be recomputed each time a sentence should be analyzed with LSA. Given that SBERT is relatively slow at computing embeddings, we therefore cached the computed embeddings. The application therefore only computes the embedding for a given sentence only once, and reuses that embedding for later analyses.

The sentences consist of words in their original word form, that is, the words are not lemmatized before being sent SBERT.

## 3.4 Implementing LSA Similarity

The LSA indexes LSASS1, LSASS1d, LSASSp, and LSASSpd are similarity measures. We implemented these indexes according to their descriptions in Section 2.5.1.1, with the only exception being the use of SBERT embeddings (see Section 3.3). We used the Numpy package for Python to do the calculations (Harris et al., 2020).

## 3.5 Implementing LSA Givenness

To compute the givenness of a sentence (see Section 2.5.1.2), we computed its embedding as well as the embeddings of its previous sentences. Then we constructed a hyperplane from the previous embeddings using a modified Gram-Schmidt process. The givenness was then computed from the projection of the embedding of the current sentence onto the hyperplane.

### 3.5.1 Simplified Gram-Schmidt for constructing the hyperplane

To construct the hyperplane of previous embeddings we used the Gram-Schmidt orthogonalization process (see Section 2.8). Gram-Schmidt is suitable for constructing an ON-basis from a set of vectors. However, we only needed it for orthogonalizing the set of vectors making up the hyperplane, as well as reducing any linearly dependent vectors to the null vector. The vectors must be orthogonal since they each represent information. By orthogonalizing them we make sure that there is no overlap of information; no two vectors contain the same information. We therefore simplified the Gram-Schmidt process to this purpose, by removing the normalization step (see step 2 in Section 2.8). Normalization, which is to turn the length of a vector into 1, is not necessary since we are only concerned with the direction of the vector and not its length. The length of the vector being projected onto is irrelevant.

Our simplified process can be described with the following steps:

1. Select the first vector from the set of vectors.
2. Fill out the hyperplane by adding the vector to it.
3. Select the next vector.
4. Project the vector onto the vectors specifying the so far constructed hyperplane.
5. Subtract the projection from the vector.

This results in a set of vectors, which can contain either non-null vectors or null vectors. The non-null vectors are orthogonal with each other, and span the hyperplane. Any resulting null vectors are of no concern, since the vectors are only used for projecting onto. A projection onto the null vector results in the null vector, and we thereby end up subtracting nothing.

### 3.5.2 Projecting onto the hyperplane and computing the indexes

The current sentence embedding was then projected onto each vector in the hyperplane (see Equation 2.9, and 2.10). The projections were summed together, resulting in a givenness vector (see Equation 2.8). As Equation 2.11 illustrates, the givenness vector was subtracted from the current sentence embedding which resulted in a newness vector. The lengths of these vectors were used for computing the proportion between given and new information, as seen in Equation 2.12. This proportion is the givenness of a sentence.

The above process was done for each sentence (except the first one) and their respective hyperplanes. The mean and standard deviation of the resulting proportions were then calculated with Equation 2.13 and Equation 2.14, which constitutes the LSAGN and LSAGNd respectively.

## 3.6 Implementing L2 Reading Index

As Section 2.5.2 states, the L2-index is an aggregation of three indexes: content word overlap (CRFCWO1), syntactic similarity (SYNSTRUTa), and CELEX word frequency (WRDFRQmc).

### 3.6.1 Content word overlap

The CRFCWO1 index measures how often content words overlap between adjacent sentences. This was computed according to Section 2.5.2.1. The POS-tag of each parsed word was used to determine whether it was a content word. Table 3.1 contains the POS-tags that constitute content words.

Table 3.1: The POS-tags and respective word classes that constitute content words (also known as open word classes). This list was provided by Universal Dependencies (2014-2022).

POS-tag	Word class
ADJ	Adjective
ADV	Adverb
INTJ	Interjection
NOUN	Noun
PROPN	Proper Noun
VERB	Verb

### 3.6.2 Syntactic similarity

The Sentence Syntax Similarity index (SYNSTRUTa) measures the mean syntactic similarity between adjacent sentences, and is described in Section 2.5.2.2. We had two approaches for computing this similarity. The first using dependency relations and the second using constituency trees.

#### 3.6.2.1 Syntactic similarity with dependency relations

The first approach relied on dependency relations between each word (see Section 2.9.2). As described in Section 2.5.2.2, SYNSTRUTa relies on constituency trees for representing

syntax. However, we were initially unable to use constituency trees, since Stanza at the time did not support constituency parsing, but only dependency parsing. Since both dependencies and constituencies represent the syntactic structure as a tree (however, in different ways), dependencies would suffice for measuring similarity.

### 3.6.2.2 Syntactic similarity with constituency trees

While dependency relations may suffice for measuring similarities, it still deviates from the original Coh-Metrix. We therefore used Benepar (see Section 3.2) for parsing constituency trees (see Section 2.9.1). The application therefore ended up with functionality for computing similarity based both on constituency trees and dependency relations.

### 3.6.2.3 Analyzing the trees

To analyze the trees (either constituency or dependency) we used the NetworkX library for Python, which provides tools for constructing and analyzing graphs (Hagberg et al., 2008).

For two sentences  $S_1$  and  $S_2$ , two graphs  $G_1$  and  $G_2$  were created from their syntactic trees. The common graph  $G_C$  where then constructed from the nodes that are both in  $G_1$  and  $G_2$ . Depending on  $G_1$  and  $G_2$  this common graph  $G_C$  may either be fully connected or not. A set of connected nodes is said to be a component of a graph. A graph may contain several components. A fully connected graph, however, contains only one component, which is the full set of nodes in the graph. We are only interested in the nodes that are connected to the root node, in other words, the component that contains the root node.  $\#NodesCommon(S_1, S_2)$  in Equation 2.19 is the number of nodes in this component.

To represent a node in a graph, we used an n-tuple containing the path of syntactic labels from the root to the current node. For constituency trees this path consists of the constituency labels from the current constituency to the root constituency. For example, a noun (NN) inside a verb phrase (VP) inside a sentence (S), would result in the tuple (NN, VP, S, TOP). For dependency trees the dependency relation labels were used instead. Using tuples on this form allows for quickly comparing the similarity of two nodes in the two graphs  $G_1$  and  $G_2$ . If two nodes have identical tuples, then they are syntactically similar, and should be added to the common graph  $G_C$ .

One drawback of this method is that, in a given sentence, a path may not be unique. For example, assuming the grammar allows it, a verb phrase may contain two nouns, resulting in two identical tuples (NN, VP, S, TOP) in the same sentence. Graphs cannot contain duplicate nodes, so one of the tuples will be excluded. In fact, it will not only exclude the specific duplicate node, but also all its sub-constituency nodes. In cases like these, the constructed graphs will be smaller than the actual syntactic tree. Our method will therefore not give a fully accurate measurement in all cases. It may occasionally underestimate the syntactic similarity between two sentences.

### 3.6.3 CELEX word frequency

The CELEX word frequency index measures the occurrence of words in the English language. To implement the WRDFRQmc we followed the procedure in Section 2.5.2.3, but with one major deviation. The CELEX word frequency index is adapted for the English language, but our index, however, was intended to measure Swedish texts. We therefore, instead of using the CELEX, used the NyLLeX corpus (Holmer and Rennes, 2022). The NyLLeX corpus contains lemmas, their POS-tags, and their word count frequencies.

In our implementation, to determine the frequency of a word, we used both the lemma and POS-tag. Using the POS-tag in addition to the lemma reduces the risk of ambiguity. A lemma may have different word senses. For example, “lead” can either be an action or a metal, and therefore have two senses. To disambiguate these, we can look at their POS-tags: “lead” as an action is a VERB, and “lead” as a metal is a NOUN.

The WRDFRQmc index only looks at content words. We therefore used the same content word list as for the content word overlap index (see Section 3.6.1).

## 3.7 Putting the indexes together

The application scores a given text using the implemented indexes LSASS1, LSASS1d, LSAGN, LSAGNd, SYNSTRUTa, and CRFCWO1. The results of these indexes are averaged together to give a final aggregate cohesion score for the text. The aggregate score for a given text is given by the following formula:

$$score = \frac{1}{6} (LSASS1 + LSASS1d + LSAGN + LSAGNd + SYNSTRUTa + CRFCWO1) \quad (3.1)$$

We ran some technical tests on whether the indexes were appropriately weighted in the formula, i.e to check if some specific indexes had better chances of predicting better orders alone than in combination with all the other indexes. We performed some tests on ChatGPT-4 (OpenAI, 2023) generated texts of a fixed length and appropriate topic and ran them through the model with different weight combinations. We evaluated the weight combinations based on how similar the model output was to the original summary. The tests therefore operated under the assumption that the sample summary outputs were very good in their original form which we assumed were a fairly reasonable hypothesis.

When running the tests according to the above assumption we found that the “best” order actually were those weighted with only LSASS and LSAGN. Also the SYNSTRUTa and LSASS1d did not seem to affect the quality of the orders at all. For the final application, however, we still chose to go with an equal weighted sum of all the indexes because the tests still proved this to be a fairly good weight combination and we did not want to deviate too much from the original instructions we got if not necessary.

It should also be mentioned that CELEX Word Frequency was excluded from the formula because it is not affected by the order of sentences, hence making the comparison of it useless.

### 3.8 Reordering sentences

To find a good order of sentences, we represented it as an optimization problem, where each search state is a particular order of sentence. To approximate a solution to this problem within a reasonable time, we employed simulated annealing (see Section 2.7). This was preferable to a brute-force approach, since the number of search states grows as  $n!$  where  $n$  is the number of sentences. This would result in a combinatorial explosion, making the application unsuitable for longer texts. This problem was mitigated with simulated annealing, which finds an approximate solution in a shorter time.

We used the Py-Search package for Python, which provides an implementation of simulated annealing (MacLellan, 2019). This requires specifying a random successor function for a given state, as well as a scoring function for a given state. The random successor function selects a random neighboring state of a given state. We did this by swapping the positions of two randomly selected sentences in the current sentence order (see Figure 3.3). The scoring function is used for determining if a state is better than another; in our case, whether a sentence order is better than another. We did this using the aggregate in Section 3.7.

Since simulated annealing only approximates the solutions, it will not always give the best sentence order, but only an order that is close to best. Further, since it also relies on randomness, it will not always give the same order on different iterations. Two different iterations may result in different orders despite having identical initial orders.

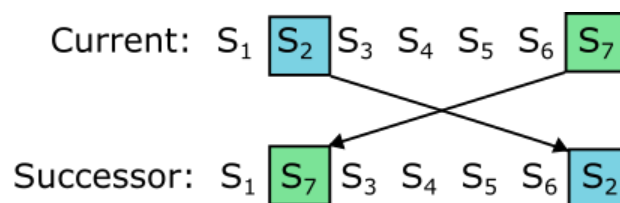


Figure 3.3: A successor state is generated by randomly swapping two sentences in the current order. In this example the sentences  $S_2$  and  $S_7$  swap positions.

## 4. Research design and procedure

This chapter describes the research design, how data was collected through the use of a survey, and how this was subsequently examined. It details the materials, participants, and ethical considerations involved, as well as the general procedure that was adhered to. The method of analyzing collected data is explained, including the statistical tests employed.

### 4.1 Design

The study utilized a within-group comparative design to evaluate the effectiveness of two different iterations of an extractive text summarizer, differing in their implementation of cohesion measures, to determine which of these was more successful in terms of readability, and subsequently what effect the modifications of cohesion measures had on the readability. The centerpiece of the study is a survey.

#### 4.1.1 Materials

To conduct our study we made use of a number of tools and materials. We summarized news articles with ElsaSum and reordered them using CohSort. We compared and evaluated them using SAPIS and an online survey. These tools and materials are described in detail below.

##### 4.1.1.1 Articles to be summarized

The articles used in the survey and the technical evaluation come from the Swedish newspaper *Dagens Nyheter* and were retrieved using the database Mediearkivet. We found 15 articles from May 2023 within the 300 - 400 word range and selected three that we considered to be unbiased and unlikely to generate strong emotions for the participants. Article 1 is about night active birds and their singing, Article 2 is about participation in culture, and Article 3 is about the use of digital tools in Swedish schools. All 15 articles were used in the technical evaluation. The short summary in the beginning of each article was excluded.

##### 4.1.1.2 ElsaSum and CohSort

ElsaSum, as detailed in Section 2.4, is an extractive text summarizer provided to us by TextAD, which utilizes various cohesion measures. CohSort, as detailed in Chapter 3, is the application developed for this study, a modified version of ElsaSum that utilizes modified and new implementations of cohesion measures. ElsaSum was used to summarize the articles outlined in Section 4.1.1.1, and CohSort was used to generate modified, reordered versions of these summaries. This served as the precursor for using these in the technical evaluation and for displaying them in the survey used in data collection.

### 4.1.1.3 Survey

The survey (see Appendix E) was conducted with six different summaries arranged in pairs. Each pair includes a text summarized using the ElsaSum summarizer and a version of the same text summarized using our CohSort application. The survey first presents a summary, then asks the participant to evaluate this summary through a set of questions which were to be answered on a six-point scale, with the answer options shifting order across questions. These are the questions included:

1. How coherent was the text?
2. How easy was the text to read?
3. How easy was it to understand the information in the text?

After the two summaries are individually presented in this way, they are followed by a section containing both summaries at once, along with questions specifically formulated for comparing the two. It consisted of similarly categorized questions, the difference being that the answerable options were either text one, text two, or that they were performing equally. These are the questions included:

1. Which text was the easiest to read?
2. Which text was the most coherent?
3. Which text was the easiest to understand?

At the end of each section, participants were given the ability to write a short comment about how they motivated certain answers. The survey itself was divided into two different response forms, differing in the order in which the summaries were presented.

The answer options in the survey were also of a varying order so that a positive answer on one question may not be presented in the same place. This was an attempt to try to optimize the answers according to Menken and Toepoel (2023) who argue that this method can improve the reliability of participant's answers.

In addition to these two evaluation sections of the survey, participants also had to answer questions in regards to their relationship with reading as well as some demographic details, including gender and age. These are the questions included:

1. What is your educational background?
2. How often do you read longer texts?
3. How often do you read news articles?
4. How much do you like to read?

The answers from this section can also be taken into consideration while analyzing the summary evaluations. The answer options in this section vary depending on what question is asked, but there are different types of multiple option answers.

#### **4.1.1.4 SAPIS**

SAPIS was used in order to find which syntactic metrics aside from LSA and L2 were changed between the ElsaSum summary of an article and the respective CohSort summary. When using SAPIS (See section 2.10), the ElsaSum summary was sent in along with the CohSort summaries and the resulting keys in the library within the JSON object returned were reviewed to find any keys with changed values. Those keys were then ordered in a matrix according to most occurring to least, where the most occurring are the metrics changed in the most comparisons.

#### **4.1.2 Participants**

Out of the 22 participants that contributed to the survey, 18 of them were currently studying at university or college, three had completed their studies at university or collage and one had a completed highschool education. These participants were recruited through convenient sampling and knew some or one of us in the project group. Approximately 41% of the participants identify as female and the rest as males. The average age of the people involved are approximately 24.5 years old. The vast majority of these participants evaluated themselves to having a positive interest in reading which does not correspond to TextAD's target group, but since this study simply focuses on reading coherence this did not defeat the purpose of the project itself.

#### **4.1.3 Ethical considerations**

The participants in the survey were informed about the project's purpose and how their involvement would be used in the study before they proceeded with the survey. A consent form (see Appendix E) was presented to every person who took part in answering the survey in which they agreed to how their information would be stored and used in the study. All information that was gathered is stored on a personal harddrive that belongs to one of us in the project group for a maximum of six months before being deleted. The participants have full anonymity and did not have to give out any personal information about themselves more than gender, age and educational background so that no answer could be traced back to a single person. This confidentiality agreement was presented to the participants before taking part of the survey so that it could be ensured that the participants had consented to being a part of this project.

### **4.2 Procedure**

CohSort was developed and implemented as a modified iteration of ElsaSum (see Chapter 3). Moving on from this, suitable articles were selected (see Section 4.1.1.1), which were subsequently summarized using ElsaSum. Each summary was then reordered using CohSort (see Section 4.1.1.2). We therefore ended up with two sets of summaries, one set of the original, unordered summaries, and a second set of reordered summaries. Once the summaries were generated, some of them were integrated into the survey (see Section 4.1.1.3). Central to

the survey was comparison and evaluation of the respective summaries generated by ElsaSum and CohSort. The survey was formulated in such a way that it would enable the participants to critically evaluate and contrast various factors related to readability between the different summaries. See figure 4.1 for an illustration of the full procedure.

The survey was sent out through two separate response forms (see Section 4.1.1.3) via various messaging services, to a group of participants (see Section 4.1.2). In order to get an even distribution of both surveys across participants, half of the project group distributed the first one and the other half the second.. It stayed active for 4 days before being closed from further data collection. We then proceeded to analyze the data through statistics and technical evaluation.

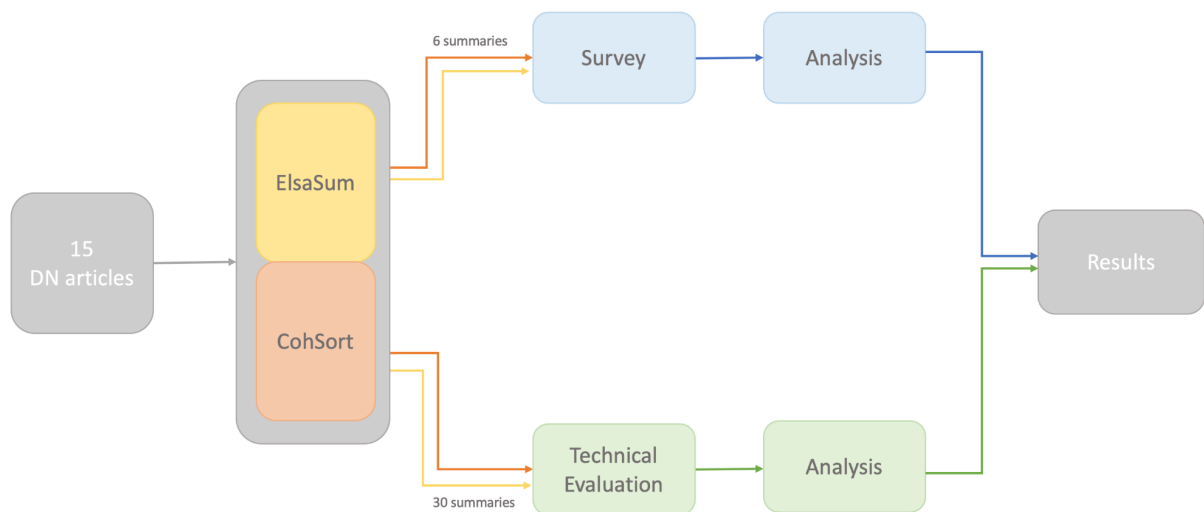


Figure 4.1: Overview of the design and procedure of the project. 15 articles were summarized resulting in a total of 30 summaries, all of which were subjected to the technical evaluation, and six of which were used in the survey.

## 4.2.1 Analysis

The analysis of data collected from the survey focused on three dimensions. These included the participants perceived coherence, ease of reading, and ease of understanding information (information gain) in the text. The participants scored the coherence and information gain dimensions on a scale of 1 to 6, with 1 being the least and 6 being the most easy to read. However, for reading ease measurement this scale was reversed, with 1 being the most and 6 being the least easy to read. For consistency, the reading ease data was reversed through a transformation, and is therefore labeled as "Reading ease ... trans" in the resulting graphs (see Chapter 5).

Additionally, a specific portion of the survey aimed to directly compare the two summaries, and the subsequent analysis involved examining the generated pie charts. This was done in order to scale the importance of the scores given the summaries without having to compare them.

The overall effectiveness of the ElsaSum and CohSort summaries was calculated by computing the mean of the three dimensions (coherence, reading ease and information ease) for each set of summaries. This provided an aggregated readability score for both the original ElsaSum summaries, and their Coh-Sort modified counterparts (see Section 5.4).

The readability mean was then further analyzed using a paired sample t-test to find any statistically significant differences between the ElsaSum summaries and the CohSort summaries. A paired sample t-test was used as the participant group stayed the same throughout all the data used. Skewness and kurtosis was measured for understanding deviations from normal distribution.

## 4.2.2 Technical evaluation

The technical evaluation of the differences between summaries generated by ElsaSum and those generated by CohSort was done using 15 different articles (see Section 4.1.1.1). Each article was summarized using ElsaSum and CohSort respectively, with the two resulting summaries subsequently being subjected to analysis using SAPIS. From the resulting JSON objects of each summaries, the included dictionaries were compared to find any changed keys, where each key represents a changed metric.

The keys were ranked in descending order of most frequently changed to least frequently changed, and cataloged in a spreadsheet. From there, the eight most frequently changed keys were chosen for further analysis, as they were changed in ten or more comparisons. This is displayed in Table 4.1, where the first column displays the name of the metric and the second displays the number of comparisons in which the metric changed.

Table 4.1: Ranking of metrics based on frequency of changes, from highest to lowest.

Changed key	No. times changed
Coh-Metrix, LSA - Adjacent sentences: Average	12
Coh-Metrix, Cohesion - Adjacent sentences, Content Words: Standard deviation	12
Coh-Metrix, Cohesion - Global, Anaphors	12
Coh-Metrix, LSA - Adjacent Sentences: Standard deviation	12
Coh-Metrix, Cohesion - Adjacent Sentences, Content Words: Ratio	12
Coh-Metrix, Cohesion - Adjacent Sentences, Stems	11
Coh-Metrix, Cohesion - Adjacent Sentences, Arguments	11
Coh-Metrix, Cohesion - Global, Content Words: Ratio	10

The metrics primarily changed stem from the Coh-Metrix tool (see Section 2.2.2). Some short summaries of the eight metrics analyzed, based on the Coh-Metrix 3.0 indices (University of Memphis, n.d.), along with their indexes within the tools' 3.0 iteration indices, are presented in Table 4.2.

As many of these metrics are measured in different ways, where some are binary scores averaged and others are standard deviations, etc. they were compared numerically but the percentage difference was also used. This was done in order to get the relative change between the ElsaSum summary and the CohSort summary. Both the numerical difference and the percentage difference were based on the mean value for the ElsaSum summary for each metric. The numerical difference was calculated by subtracting the value for the CohSort summary from the value of the ElsaSum value and the percent difference was calculated using the following formula:

$$\text{Percent Difference} = \frac{|V1 - V2|}{(V1 + V2)/2} \times 100 \quad (4.1)$$

Where the V1 value in the formula was the metrics value for the ElsaSum summary, and V2 the metrics value for the CohSort value.

Table 4.2: Description of metrics used.

Index	Metric	Description
40	Coh-Metrix, LSA - Adjacent sentences: Average	Measures the conceptual similarity between adjacent sentences by computing the LSA cosines. A higher score means a larger conceptual similarity between the sentences.
35	Coh-Metrix, Cohesion - Adjacent sentences, Content Words: Standard deviation	Measures the standard deviation of the proportion between overlapping content words within adjacent sentences.
39	Coh-Metrix, Cohesion - Global, Anaphors	Measures anaphors in the text as a whole by giving each sentence a binary score and then getting the average of those scores for all the sentences to get the global score. A higher score indicates a larger proportion of overlapping words within the sentences.
41	Coh-Metrix, LSA - Adjacent Sentences: Standard deviation	Measures the standard deviation of LSA cosines for adjacent sentences in order to measure how semantically overlapping they are.
34	Coh-Metrix, Cohesion - Adjacent Sentences, Content Words: Ratio	Measures the ratio of the proportion of overlapping content words within two adjacent sentences. A higher score indicates a larger proportion of overlapping words within the sentences.
30	Coh-Metrix, Cohesion - Adjacent Sentences, Stems	Measures the overlap between a noun in one sentence and a content word with the same stem in the previous sentence in two adjacent sentences.
29	Coh-Metrix, Cohesion - Adjacent Sentences, Arguments	Measures the overlap between a noun in one sentence and a pronoun in another adjacent sentence.
36	Coh-Metrix, Cohesion - Global, Content Words: Ratio	Measures the global ratio of overlapping content words. A higher score indicates more overlapping content words between all the sentences.

## 5. Results

This chapter presents the findings generated from the study, providing an objective account of both the survey results and the technical evaluation of the CohSort application. All the summaries that were used to garner results can be found in Appendix C.

### 5.1 Survey article 1 summary

There was a significant difference found between the coherency of the ElsaSum summary of Article 1 (mean: 4.38, SD = 1.12) and the coherency of the CohSort summary of Article 1 (mean: 3.33, SD = 1.24), as shown by a paired samples t-test,  $t(20) = 3.28$ ,  $p = 0.004$ .

There was no significant difference found between the ElsaSum summary and the CohSort summary of Article 1 in reading ease (*ElsaSum*: mean = 3.90, SD = 1.34, *CohSort*: mean = 3.0, SD = 1.30), as shown by a paired samples t test,  $t(20) = 2.08$ ,  $p = 0.051$ . There was also no significant difference found in information ease (*ElsaSum*: mean = 4.33, SD = 1.32, *CohSort*: mean = 3.71, SD = 1.31), also shown by a paired samples t-test,  $t(20) = 1.550$ ,  $p = 0.137$ .

The skewness of the three measurements for the ElsaSum summary of Article 1 (Coherency: -0.146, Reading ease: -0.225, Information ease: -0.544) and its kurtosis (Coherency: -0.409, Reading ease: -0.0885, Information ease: -0.641) shows it is a mesokurtic curve without a significant deviation from the normal distribution. The same stands true for the CohSort summary's skewness (Coherency: 0.336, Reading ease: 0.449, Information ease: 0.145) and kurtosis (Coherency: -0.151, Reading ease: -0.190, Information ease: -0.986).

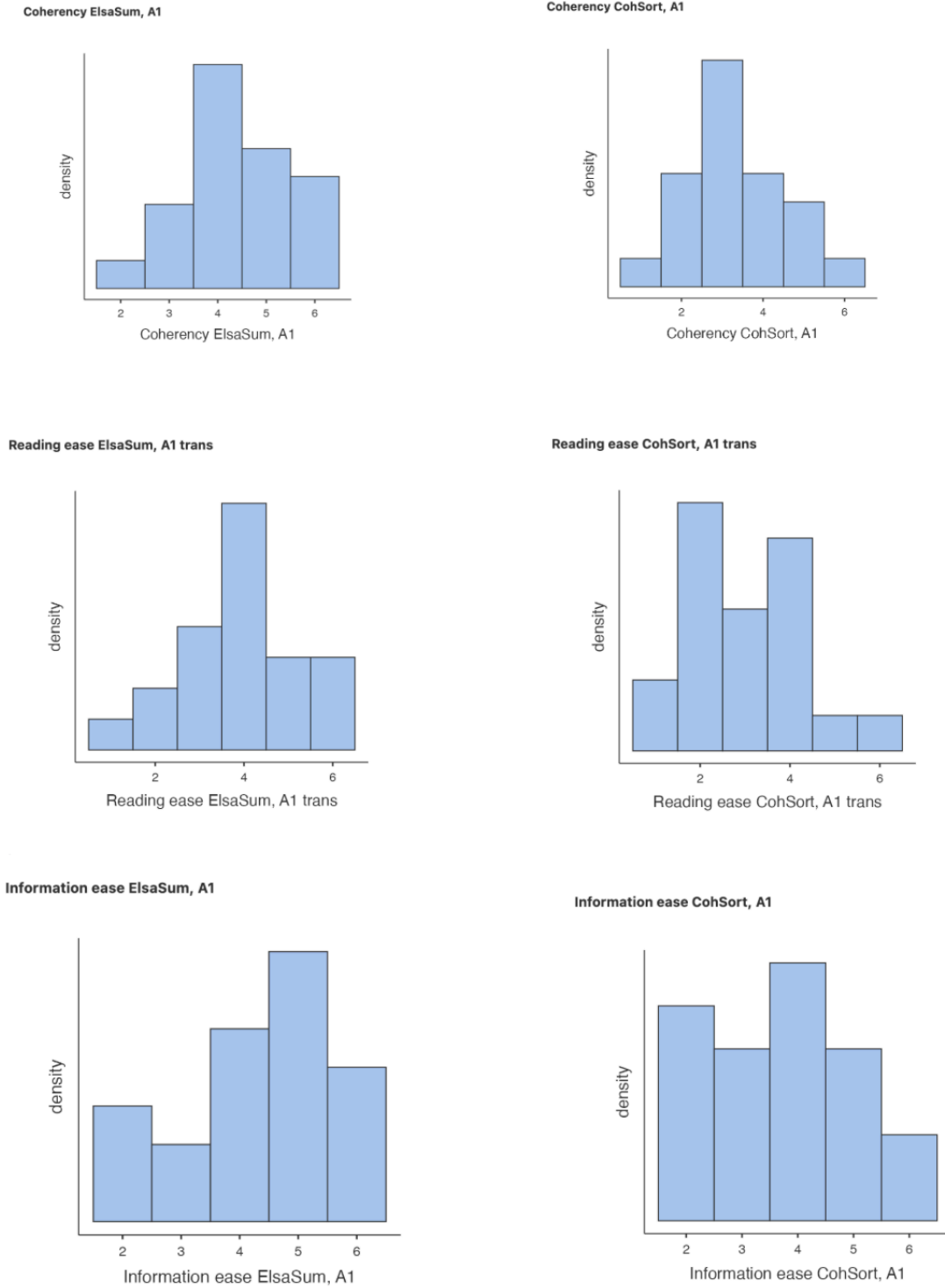
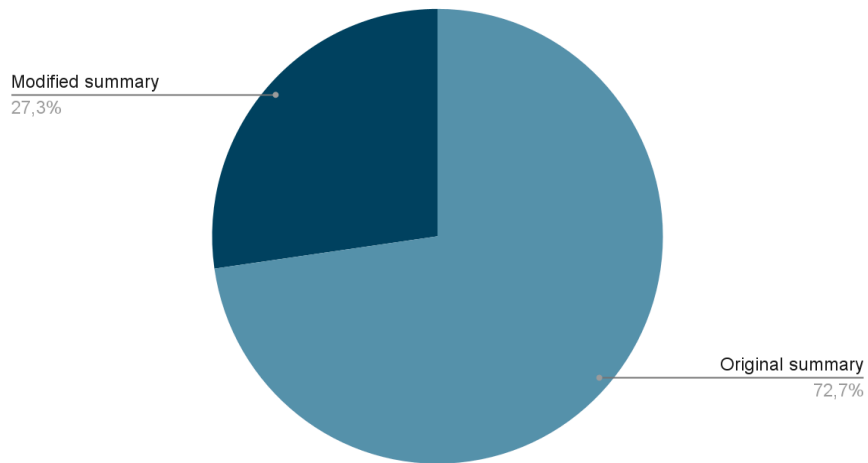
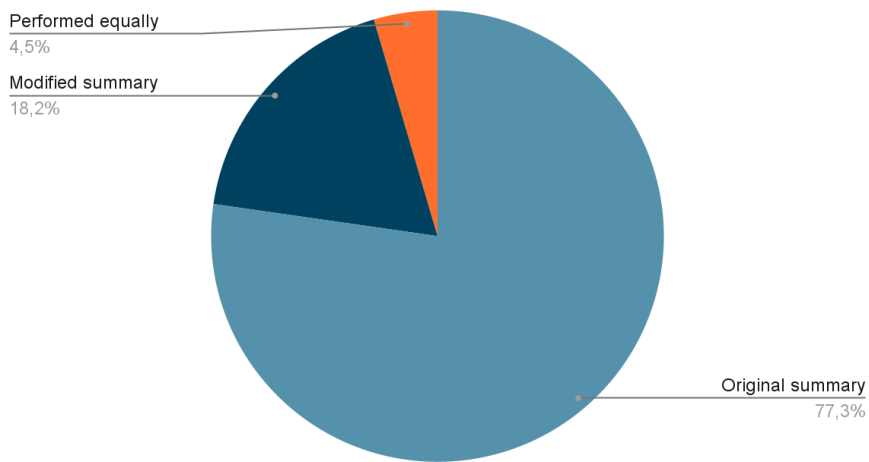


Figure 5.1: A matrix of all the histograms showing the distribution of data from the summaries of Article 1. The rows correspond to each of the three measurements (Coherency, Reading ease and Information ease). The first column displays the ElsaSum summary of Article 1 and the second column displays the CohSort summary of Article 1.

Comparison 1: Which text was the easiest to read?



Comparison 1: Which text was the most coherent?



Comparison 1: Which text was the easiest to understand?

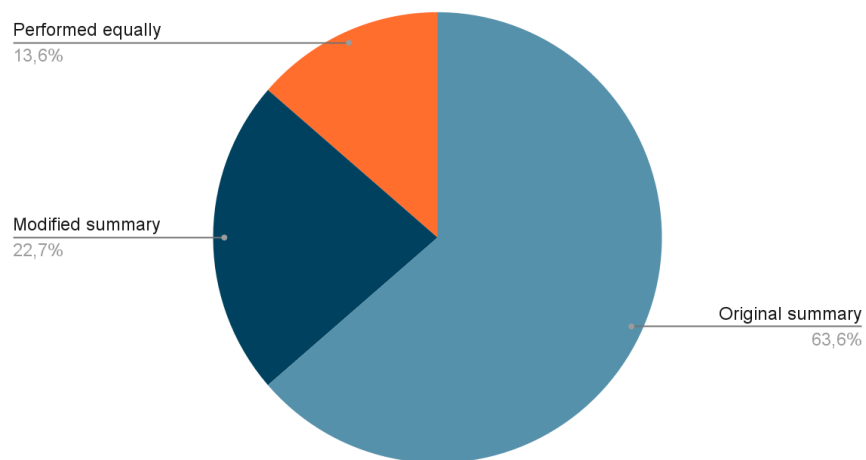


Figure 5.2: The collected circle diagrams from the comparison of Article 1. The charts show the participants' responses to the questions in the first comparison section of the survey that compares the original summary (ElsaSum) with its modified counterpart (CohSort).

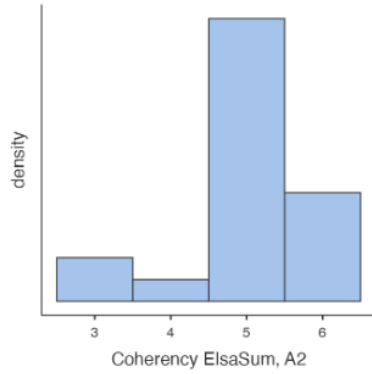
## 5.2 Survey article 2 summary

There was a significant difference found between the coherency of the ElsaSum summary of Article 2 (mean = 5.00, SD = 0.837) and the coherency of the CohSort summary of Article 2 (mean = 4.240, SD = 1.221), as shown by a paired samples t-test,  $t(20) = 3.200$ ,  $p = 0.004$ . There was also a significant difference found between information ease of the ElsaSum summary of Article 2 (mean = 4.90, SD = 0.831) and the CohSort summary of Article 2 (mean = 4.24, SD = 1.411), also shown by a paired samples t-test,  $t(20) = 2.256$ ,  $p = 0.035$ .

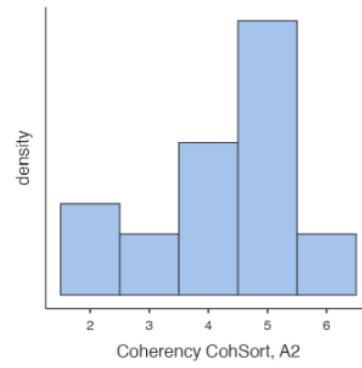
There was no significant difference found between the reading ease of the ElsaSum summary of Article 2 (mean = 3.62, SD = 1.627) and the CohSort summary of Article 2 (mean = 3.52, SD = 1.470), as shown by a paired samples t-test,  $t(20) = 0.241$ ,  $p = 0.812$ .

The skewness of coherency for the ElsaSum summary of Article 2 (-1.13) and the kurtosis (1.73) indicates a skewed dataset that is somewhat leptokurtic. The skewness of coherency for the CohSort summary of Article 2 (-0.686) and its kurtosis (-0.339) shows it is mesokurtic and not significantly skewed. For reading ease, the skewness of both the ElsaSum summary of Article 2 (0.458) and the CohSort summary of Article 2 (0.522) does not indicate a significantly skewed dataset. However, the kurtosis for the ElsaSum (-1.16) and the CohSort (-1.12) summaries shows both datasets are leptokurtic. For information ease, the skewness for the ElsaSum summary of Article 2 (-0.389) and for the CohSort summary of Article 2 (-0.349) does not indicate skewed datasets. The kurtosis for the ElsaSum summary (-0.150) shows it is mesokurtic, however the kurtosis for the CohSort summary (-1.32) shows it is leptokurtic.

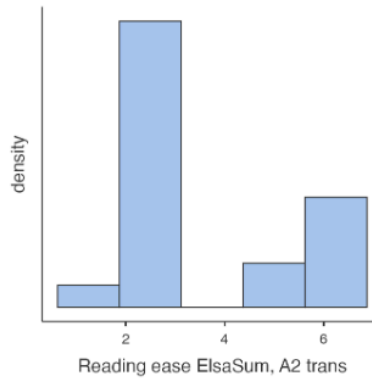
Coherency ElsaSum, A2



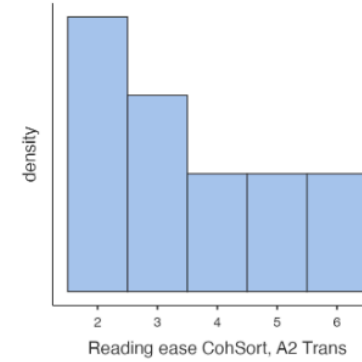
Coherency CohSort, A2



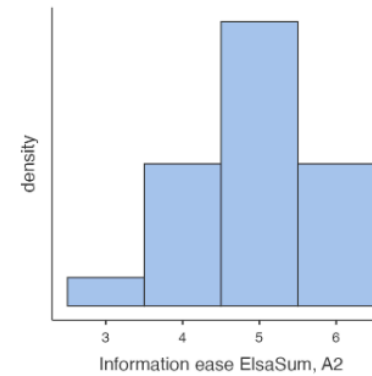
Reading ease ElsaSum, A2 trans



Reading ease CohSort, A2 Trans



Information ease ElsaSum, A2



Information ease CohSort, A2

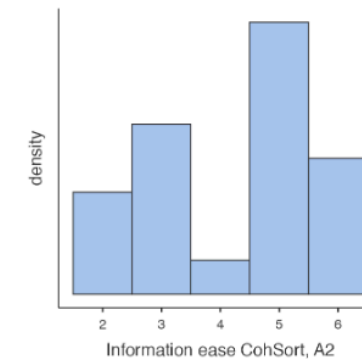
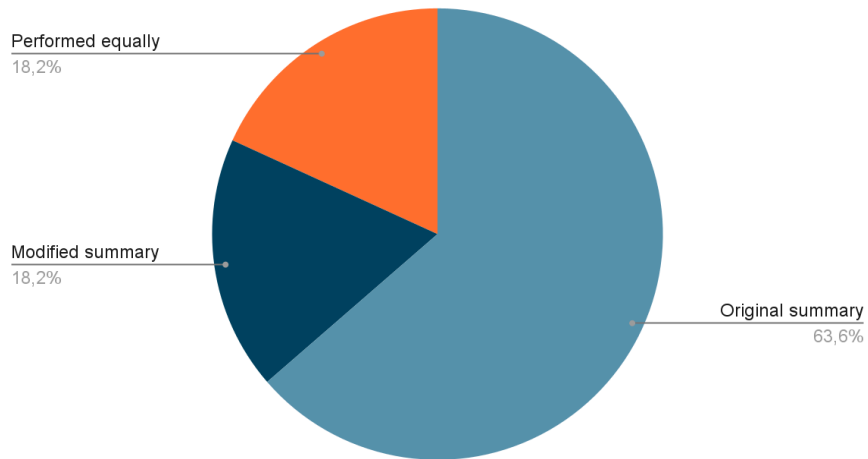
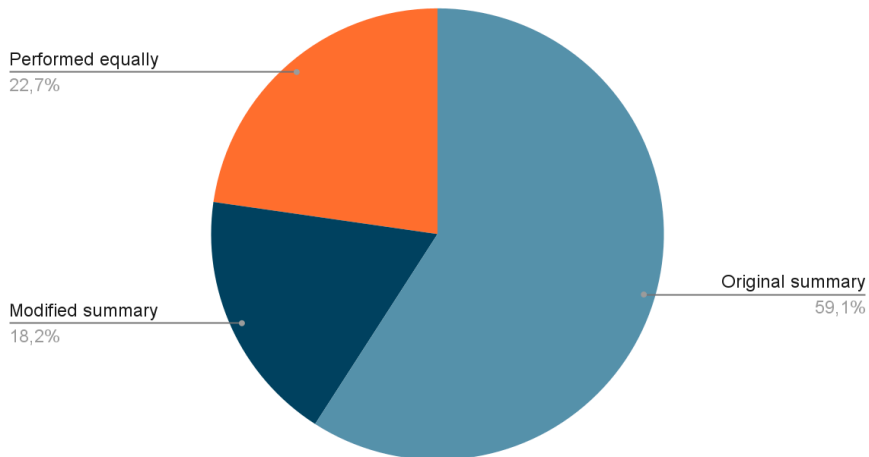


Figure 5.3: A matrix of all the histograms showing the distribution of data from the summaries of Article 2. The rows correspond to each of the three measurements (Coherency, Reading ease and Information ease). The first column displays the ElsaSum summary of Article 2 and the second column displays the CohSort summary of Article 2.

### Comparison 2: Which text was the easiest to read?



### Comparison 2: Which text was the most coherent?



### Comparison 2: Which text was the easiest to understand?

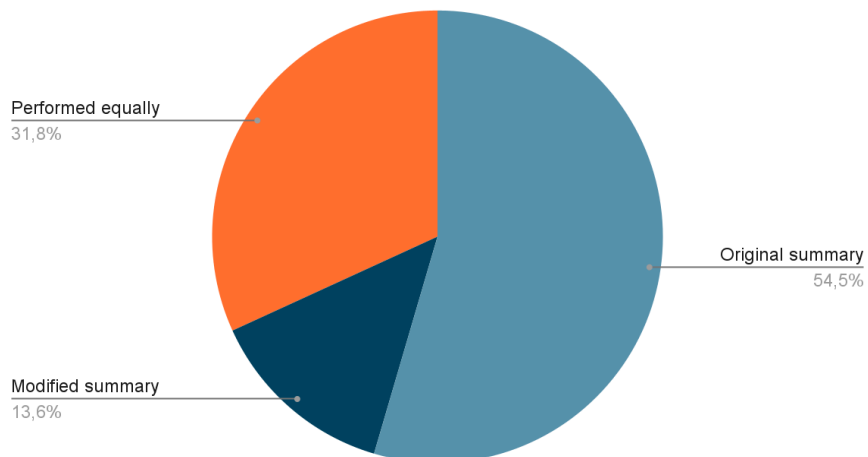


Figure 5.4: The collected circle diagrams from the comparison of Article 2. The charts show the participants' responses to the questions in the first comparison section of the survey that compares the original summary (ElsaSum) with its modified counterpart (CohSort).

### 5.3 Survey article 3 summary

There was a significant difference found between coherency of the ElsaSum summary of Article 3 (mean = 5.19, SD = 0.837) and coherency of the CohSort summary of Article 3 (mean = 4.48, SD = 1.123), as shown by a paired samples t-test,  $t(20) = 2.500$ ,  $p = 0.021$ .

There was no significant difference found between the ElsaSum summary and CohSort summary of Article 3 in reading ease (*ElsaSum*: mean = 3.90, SD = 1.972, *CohSort*: mean = 3.76, SD = 1.729), as shown by a paired samples t-test,  $t(20) = 0.420$ ,  $p = 0.679$ . There was also no significant difference found in information ease (*ElsaSum*: mean = 5.24, SD = 0.889, *CohSort*: mean = 4.71, SD = 1.231), also shown by a paired samples t-test,  $t(20) = 1.562$ ,  $p = 0.134$ .

The skewness for the ElsaSum summary and the CohSort summary show no significant skewness for coherency (*ElsaSum*: -0.902, *CohSort*: -0.168) nor for reading ease (*ElsaSum*: -0.243, *CohSort*: -0.171). For coherency, the kurtosis (*ElsaSum*: 0.332, *CohSort*: -0.335) is indicative of a mesokurtic curve, whereas for reading ease the kurtosis for the ElsaSum summary (-1.68) and the kurtosis for the CohSort summary (-1.41) show that they are leptokurtic curves.

However for information ease the skewness of the ElsaSum summary (-1.46) shows a significant skewness and the kurtosis (2.25) shows its platykurtic. For the CohSort summary neither the skewness (-0.995) or the kurtosis (0.458) show a significant deviancy from a mesokurtic normally distributed curve for information ease.

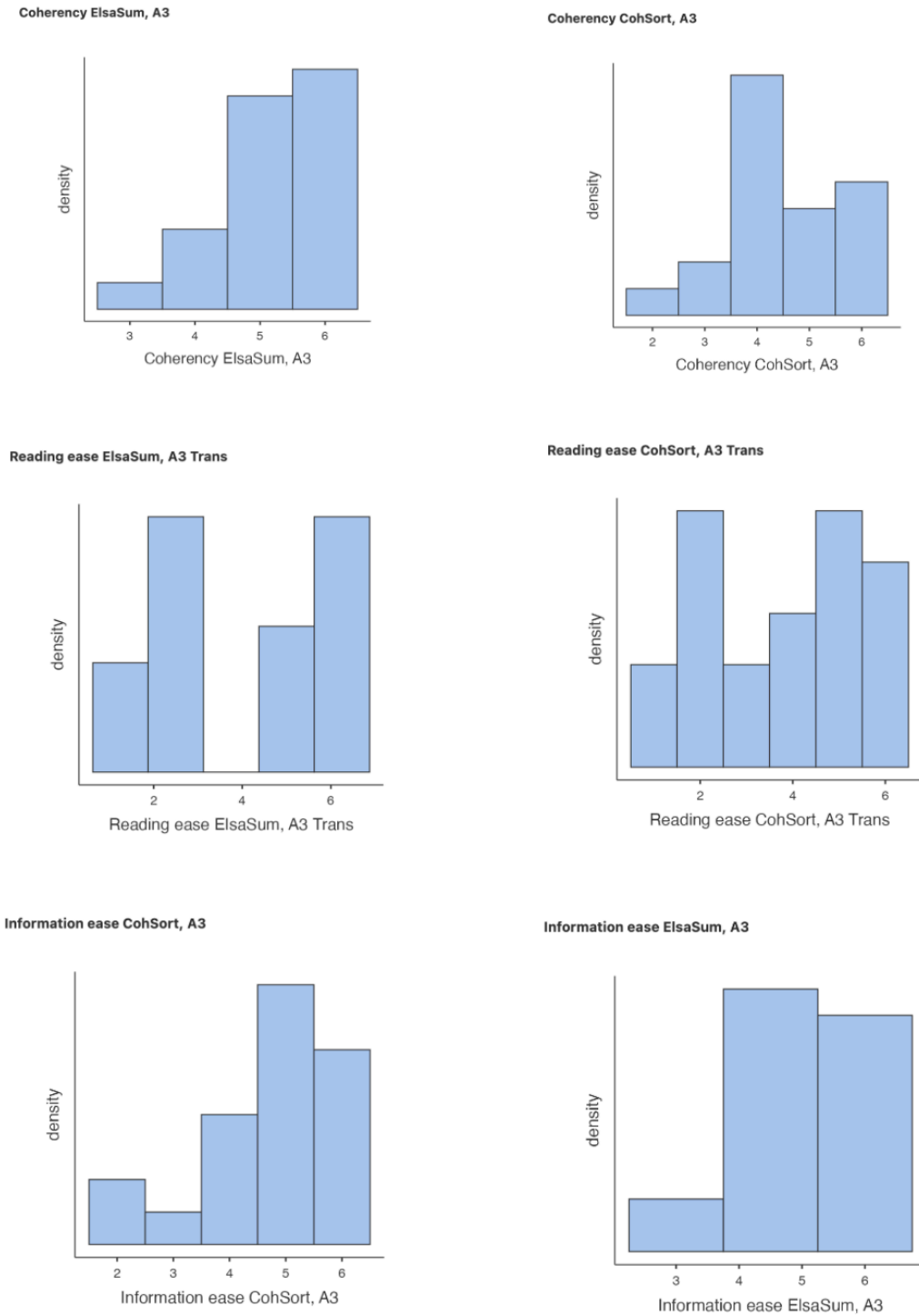
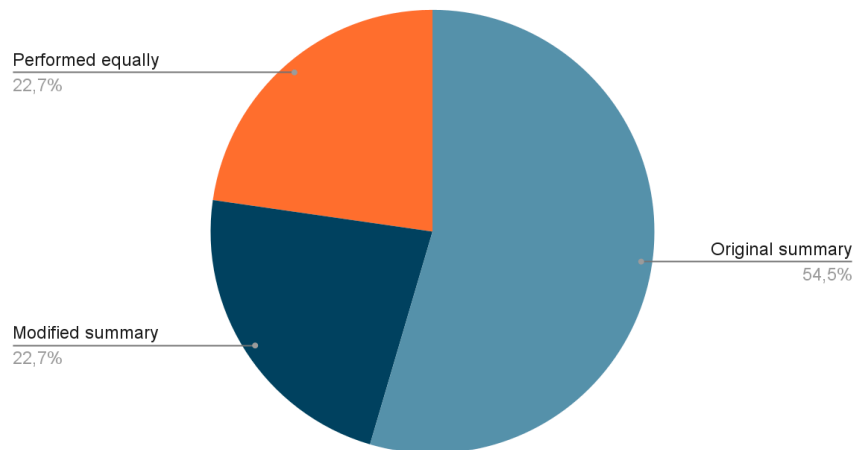
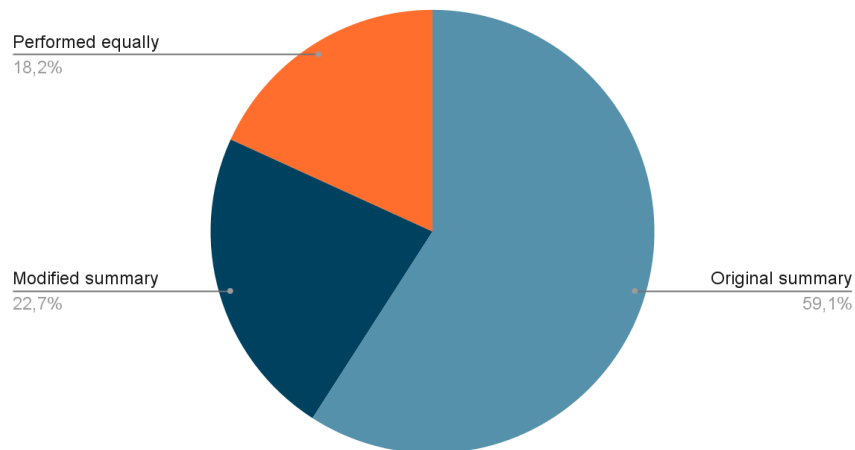


Figure 5.5: A matrix of all the histograms showing the distribution of data from the summaries of Article 3. The rows correspond to each of the three measurements (Coherency, Reading ease and Information ease). The first column displays the ElsaSum summary of Article 3 and the second column displays the CohSort summary of Article 3.

### Comparison 3: Which text was the easiest to read?



### Comparison 3: Which text was the most coherent?



### Comparison 3: Which text was the easiest to understand?

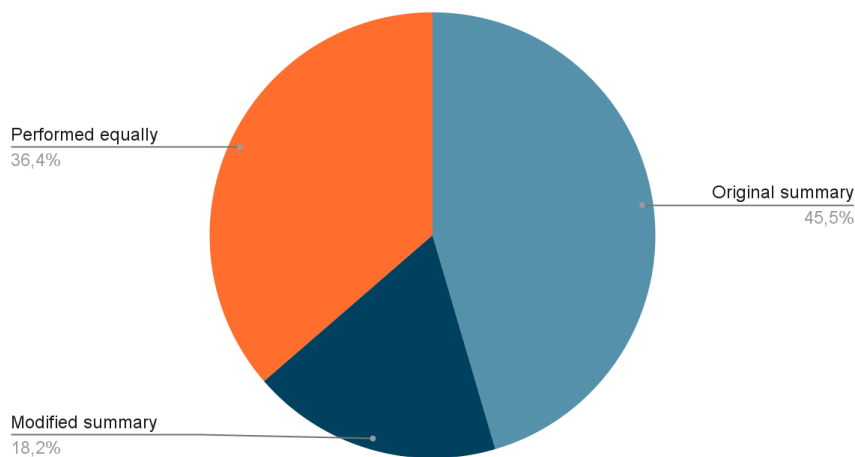


Figure 5.6: The collected circle diagrams from the comparison of Article 3. The charts show the participants' responses to the questions in the first comparison section of the survey that compares the original summary (ElsaSum) with its modified counterpart (CohSort).

## 5.4 Mean readability for each article

We computed the mean for the three measures (coherency, information ease, and reading ease), resulting in a readability mean for each article. There was a significant difference between the mean readability of the ElsaSum summary and the mean readability of the CohSort summary of Article 1 (*ElsaSum*: mean = 4.21, SD = 0.806, *CohSort*: mean = 3.35, SD = 0.980 ), as shown by a paired samples t-test,  $t(20) = 2.82$ ,  $p = 0.011$ , and of Article 2 (*ElsaSum*: mean = 4.52, SD = 0.847, *CohSort*: mean = 4.00, SD = 1.038), also shown by a paired samples t-test,  $t(20) = 2.11$ ,  $p = 0.048$ .

There was no a significant difference found between the mean of the ElsaSum summary and the mean of the CohSort summary of Article 3 (*ElsaSum*: mean = 4.78, SD = 1.002, *CohSort*: mean = 4.32, SD = 1.067), also shown by a paired samples t-test,  $t(20) = 1.65$ ,  $p = 0.114$ .

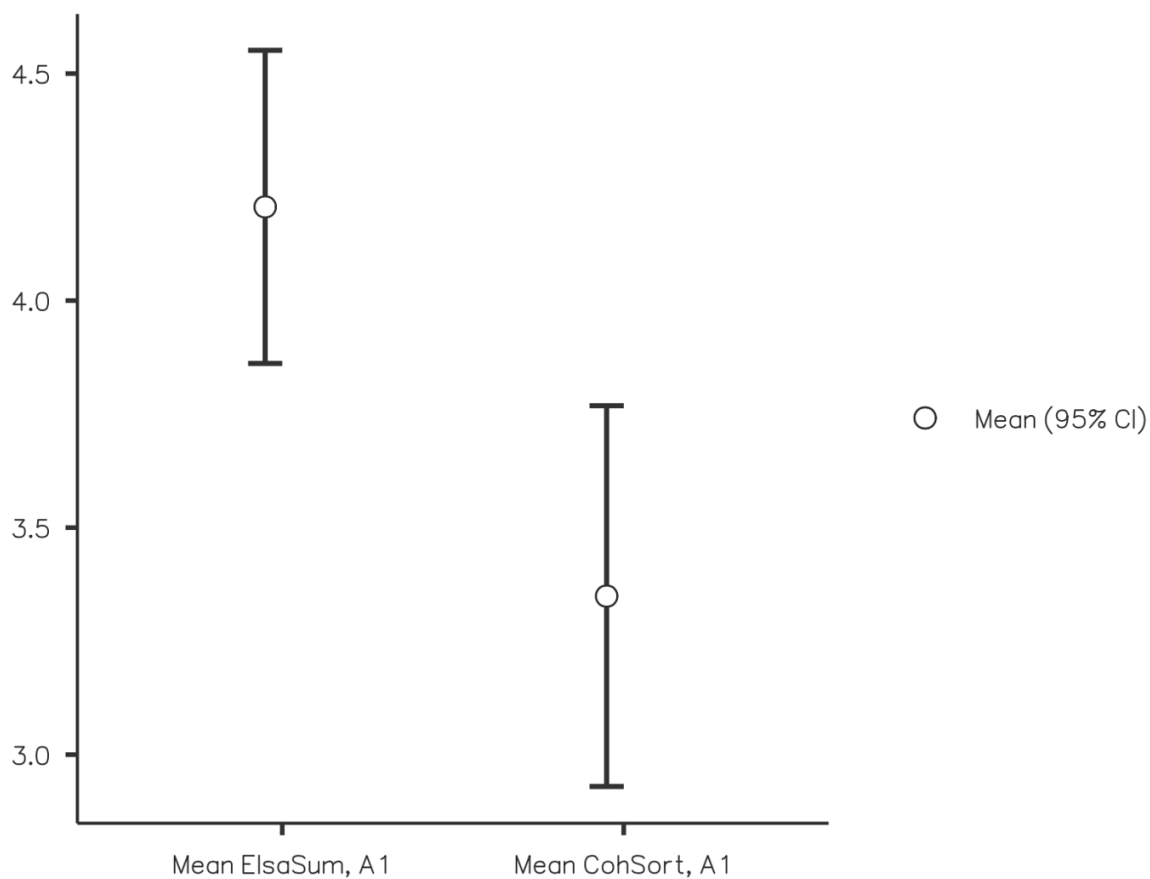


Figure 5.7: Comparison between the mean readability for the original summary of Article 1 (Mean ElsaSum, A1) and the mean readability of the modified summary of Article 1 (Mean CohSort, A1).

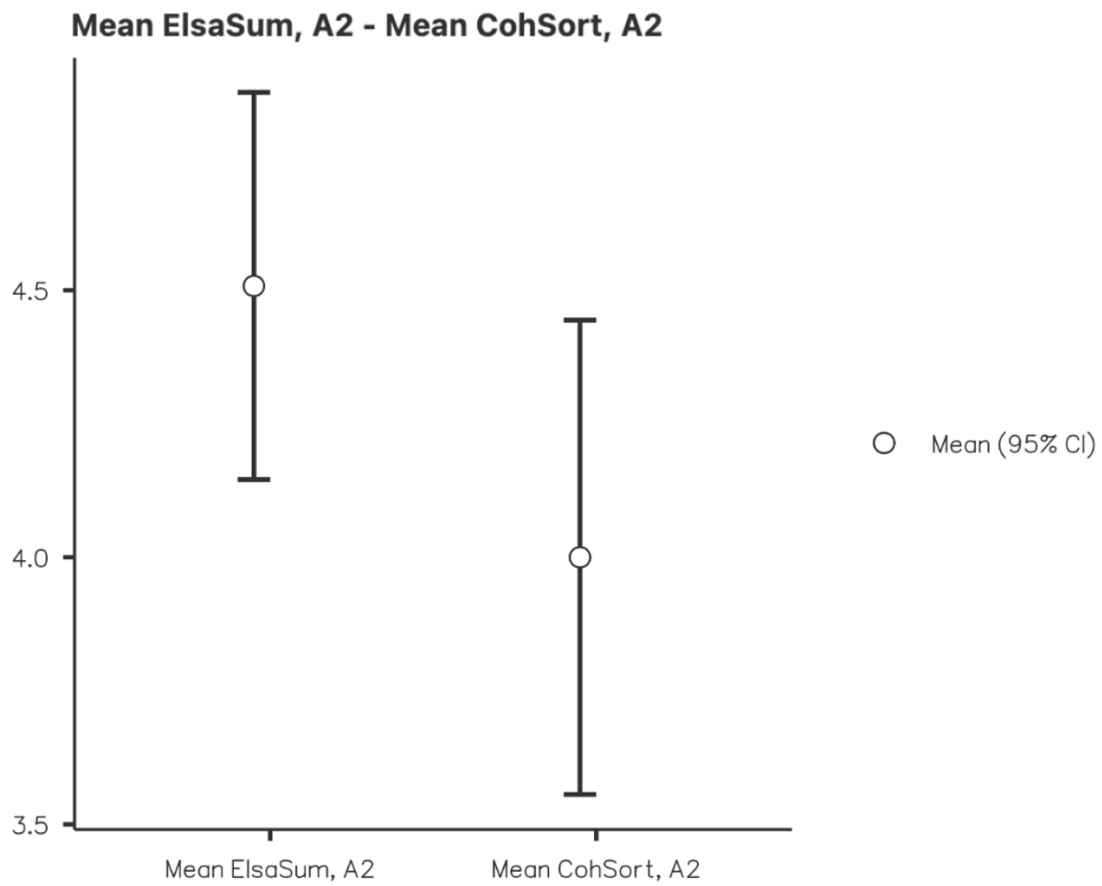


Figure 5.8: Comparison between the mean readability for the original summary of Article 2 (Mean ElsaSum, A2) and the mean readability of the modified summary of Article 2 (Mean CohSort, A2).

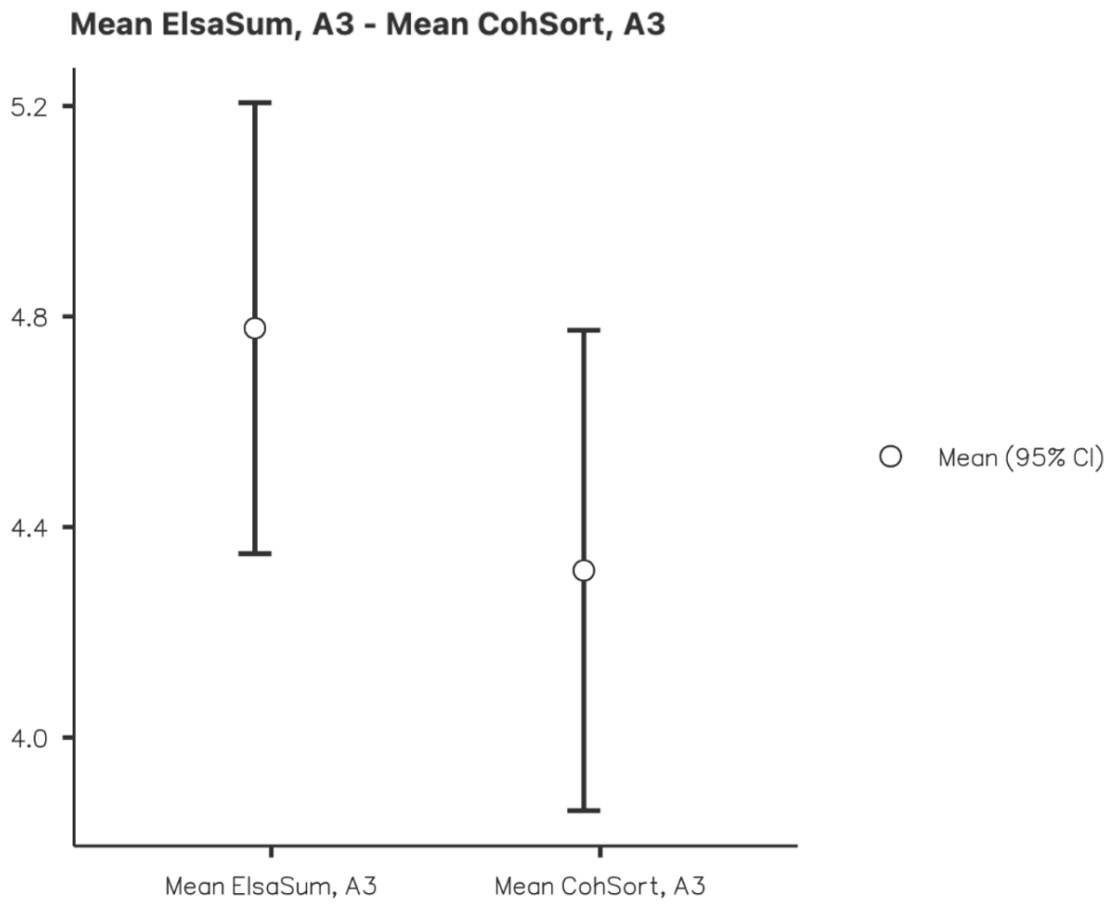


Figure 5.9: Comparison between the mean readability for the original summary of Article 3 (Mean ElsaSum, A3) and the mean readability of the modified summary of Article 3 (Mean CohSort, A3).

## 5.5 Technical evaluation

Using SAPIS, a total of 68 metrics were found to have changed between the ElsaSum summaries and the CohSort summaries (see Appendix D). The eight most commonly changed metrics are presented in table 5.1 with the metrics score for the ElsaSum summary, the metrics score for the CohSort summary and the numerical and percentage difference between them. In this table, “Changed key” represents the key changed, ordered from most often changed to least. “ElsaSum Mean” is a mean value for the metric in the ElsaSum summaries in all of the comparisons where the metric changed, and “CohSort Mean” is the equivalent for the CohSort summaries. The “Numerical Difference” column presents the numerical difference between the two metric values, and the “Percentage Difference” presents the percentage difference between the metric values.

Table 5.1: Overview of the results from the technical evaluation.

Changed key	ElsaSum Mean	CohSort Mean	Numerical Difference	Percentage Difference
Coh-Metrix, LSA - Adjacent sentences: Average	0.531112	0.540587	0.009475	1.784035%
Coh-Metrix, Cohesion - Adjacent sentences, Content Words: Standard deviation	0.135508	0.172647	0.037139	27.407258%
Coh-Metrix, Cohesion - Global, Anaphors	0.1875	0.191964	0.004464	2.380955%
Coh-Metrix, LSA - Adjacent Sentences: Standard deviation	0.105194	0.135411	0.030217	28.725022%
Coh-Metrix, Cohesion - Adjacent Sentences, Content Words: Ratio	0.011632	0.017983	0.006351	54.597927%
Coh-Metrix, Cohesion - Adjacent Sentences, Stems	0.337662	0.480519	0.142857	42.307691%
Coh-Metrix, Cohesion - Adjacent Sentences, Arguments	0.337662	0.506494	0.168831	50%
Coh-Metrix, Cohesion - Global, Content Words: Ratio	0.008872	0.008786	-0.000086	0.968915%

Percentually the largest difference was found in the metric for measuring the ratio of content words in adjacent sentences (Coh-Metrix, Cohesion - Adjacent Sentences, Content Words: Ratio) with a percentage difference of 54.59%. Whereas the smallest change was found in one

of the less frequently changed metrics (among the eight most common) “Coh-Metrix, Cohesion - Global, Content Words: Ratio” which measures the ratio of content words globally in the summary and had a percentage change of 0.97%. This was the only percentage difference in the top eight frequently changed keys which denoted a negative change from the ElsaSum value to the CohSort summary with a numerical change of -0.000086. Figure 5.10 shows an overview of the percentage differences.

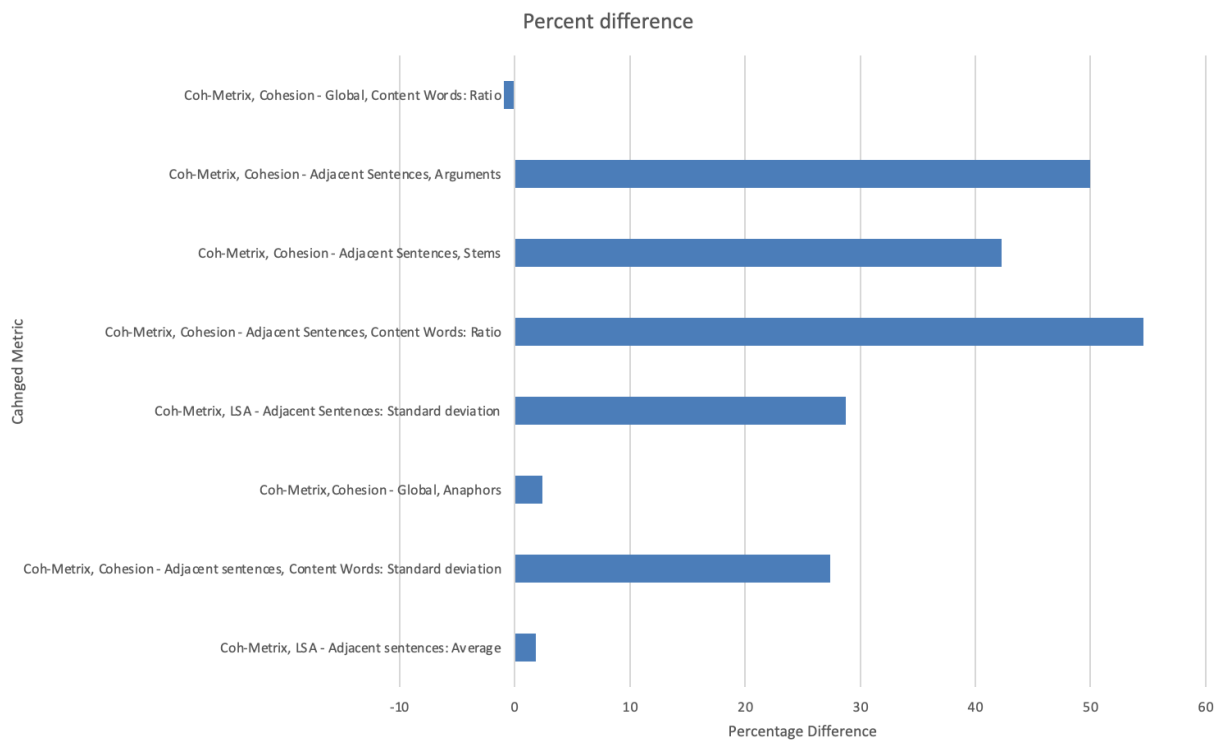


Figure 5.10: The percentage differences between the changed metrics between the ElsaSum summaries and the CohSort summaries, visualized as a grouped, laying down, bar chart. The percent are shown on the x-axis as decimals and the changed metrics on the y-axis.

The average percentual change in the top 8 (most frequently changed) metrics is a 25,779% increase in the CohSort metric values compared to ElsaSum.

## 6. Discussion

When interpreting the results of the survey the common thread is an overall higher readability score given to the ElsaSum summaries. Notably, this pattern holds true despite certain parts of the datasets displaying significant skew (see Section 5.1–5.4). Whether we look at the skewed parts or the less skewed ones, the summaries generated by ElsaSum show a consistently better score. Looking at each of the three measurements individually further reinforces this observation, where the coherence measurements had a statistically significant difference between the summaries in all cases. Within these cases there was one instance where the data was skewed, and it occurred in the ElsaSum summary of Article 2. However this could be attributed somewhat to the article being hard to summarize as it was the article with the most skewed data, while also having the most statistically significant differences between the summaries. Because of this, it is hard to draw any conclusions using Article 2, as a large part of the data gathered may be misleading, which might be why it has seemingly got so many statistically significant differences when comparing the two summaries. It is also worth noting that the result for comparing reading ease between the summaries of Article 1 was not statistically significant and is thus not handled as such in this study. However, with a p-value of 0,051 it is only one thousandth away from being considered significant, and could therefore be argued as being significant for future projects or for more practical implementations of our findings.

When comparing three measurements, which rate the summaries individually, to the results of how the summaries were rated in comparison to each other from the resulting pie-charts it is clear that there was a strong preference for the ElsaSum summary when comparing the two side by side. Using this as a starting point, the previously discussed results can be argued to be seen as significant in relation. Had we found a completely different trend in the comparison, or had the participants generally shown a low preference for any of the summaries it could have indicated that the previous results were misleading. As it stands it does the opposite and shows that the scores seem to fairly accurately reflect the preference for the ElsaSum summary which the survey participants also showed when scoring the two summaries.

When evaluating the difference between the ElsaSum summaries and the CohSort summaries using linguistic metrics through SAPIS we get another trend. Instead of finding that the ElsaSum summaries are more linguistically cohesive according to the metrics changed between the summary versions, the opposite stands true. Looking at the technical results we can see that the percentage difference varies from. In Figure 5.10 it is clear that though there are some extremes, most changes are around the 20-th percentile, which is in line with the average (25.779%).

To summarize the results it seems that even though the CohSort summaries fairly consistently rank higher than the ElsaSum summaries in computed linguistic metrics, readers do not actually find it more coherent. Following this finding it could be argued that the “Coh-Metrix, Cohesion - Global, Content Words: Ratio” metric, which was the only one with a negative

numerical difference (-0.000086) and the smallest percentage difference (0.97%), was the heaviest weighing metric. However as this is also the smallest percentage difference, with a 24.809% difference to the average this seems unlikely. It would have to have an extreme effect on the cohesion of the text for it to be the cause of the different results of the survey and the technical evaluation.

## 6.1 Survey discussion

There are some points worthy of discussion and clarification with regards to the survey conducted for the study. To begin with, the reason for using an even number of options in items of the first part of the survey was so that the participants would be forced to take a stance. If an uneven number of options were to be presented it would have given the participants the option to be neutral, which would not be relevant data for a comparison between the two summaries. Some might argue that this is not an appropriate way to gather information about a subjective and opinionated question. However, because this part of the survey is constructed to gain information about the readability of each individual summary, it could help us in a comparative analysis, and therefore the decision was made.

The reason for having separate sections for the individual summaries and the comparison was to see if there was any correlation between these two on the individual scale, meaning that a participant's answers can be analyzed in such a way that it could be seen what was causing one summary to be more preferable than the other. This section also had the option to not prefer either summary by selecting an option that says that the texts performed equally in a certain category. The reason for this is so that one participant's opinion within a field can be valued in different ways. If a participant for example gives one summary a higher score in readability than the other but is valued the same in the comparison chapter it could be analyzed how much this actually weighs into a comparison. This was done because it is otherwise hard to value a person's ability to self reflect.

The ability for participant's to include free text answers at the end of some sections in the survey served as a way to gather more insight into what the individuals thought about the section itself. This was also done so that an analysis could determine an answer's reliability without drawing unmotivated conclusions. As mentioned in the previous paragraph, it is hard to value an individual's ability to self reflect.

The reason for dividing the survey into two different response forms with different orders was to see if this had any impact on which summary participants would prefer. If there is a trend that differs between the two forms, the order could act as a factor for that trend and would then have to be taken into consideration while analyzing the survey's answers. Here we saw some effect on the results which seems to stem from the order in which summaries were presented. In the form that presented the ElsaSum summary first we noticed that participants evaluated the CohSort summary much higher than in the form that presented the CohSort summary first. This could potentially be a result of the order of the summaries, but that conclusion cannot be drawn with certainty. The difference between the participants who answered the different forms is too substantial to conclude that the order is the only potential

factor for this trend. There is not an even distribution in regard to age, gender or educational background between the two forms which makes it harder to assume what factor results in what trend. This is however an interesting trend that we wanted to illuminate to facilitate future research.

The articles we chose to summarize may also have influenced our results. No prior testing of what a summary of the specific articles would look like was done before including the summaries in the survey. Other than choosing articles that were unlikely to elicit strong emotions in our participants, there was no specific method applied when selecting them. There could possibly be a difference in how different types of articles are modified by CohSort as well as a difference in how the summaries are perceived depending on the content. This is however beyond the scope of our project.

The length of the included articles is also something that could have affected our results. One could imagine that longer articles would lead to a different outcome, since more information would have to be excluded in order to produce the summary and thereby affect the cohesiveness.

As mentioned, the survey relies on reports of self-reflection of the individual. While we have tried to reduce sources of cognitive biases in these reports, they still rely on subjective assessments. There are many other ways to measure the readability of a text. Another approach would be a reading comprehension test, where we would get a more or less objective measure of how much of the text the individual understood. A more readable text would yield a higher understanding than a less readable one. Another approach would be to measure eye fixations using eye-tracking technology, where longer fixations and rereading of sentences would suggest a reduced reading flow and therefore a less readable text. The limited resources of this project, however, made self-report surveys the most suitable option.

Another point of discussion is the method for gathering participants for the study. A total of 22 participants were recruited through convenient sampling for the study. This method ended up yielding a lot of Linköping University students in their 20's for the survey, as well as university educated participants (with only one exception). Although the sampling was helpful in limiting the differences between participants and thus potential factors to take into account, it means that the results gathered are difficult to generalize in any meaningful way.

A final point of discussion is the potential issue of flipping the scales for the reading ease survey item. We initially did this as an attempt to minimize placement biases, such as the participant preferring higher numbers compared to lower numbers. However, in hindsight we realize that this poses a potential problem. The participants may not have noticed that the scales changed on these items, and may therefore have given a higher rating when a lower one was intended, and vice versa. We have not explored to what extent this could have affected our results.

## 6.2 Technical evaluation

The technical evaluation for the project ended up being quite narrow, and although it gave some useful insights, a more thorough evaluation could have given us a deeper understanding of how the summaries differed. There are a few ways in which the technical evaluation could have been improved.

The first possible improvement for the technical evaluation would have been to evaluate a larger portion of the metrics which changed. When selecting the metrics for evaluation in this project, the time available for evaluation along with what could be relevantly evaluated played a big role. Only 8 out of 68 changed metrics were evaluated, which is a little over a tenth of the total number of changed metrics. However, because of the way they were selected, the metrics did provide insight into some of the larger trends distinguishing the summaries of ElsaSum and CohSort respectively. Additionally, removing the metric for seeing the length and order of the sentences streamlined the process of analyzing our values, due to it being handled as a list, which would have required a different analysis method compared to the other metrics. With more time available it could have been interesting to examine whether there was any tendency in how the ordering of different length sentences differed between the summaries. However, as it is a metric which was guaranteed to change in every comparison, it was not prioritized in this study. Furthermore, the most commonly changed metrics total to 92 out of 185 total changes (49.73%) in the comparisons. It is worth noting that the total of 185 is after removing the metric for sentence length order.

Another improvement would have been to include a few additional metrics related to those which demonstrated differences between the summaries. There could have been value in exploring whether there was any correlation between the global and adjacent metrics for all of the most frequently changed metrics. One example for this is the metric for measuring the stem overlap in adjacent sentences; it would have been interesting to see if differences existed in the summaries where this metric changed. Specifically, it might have been more interesting in that case to see if the metric change was due to a low global metric, making it more likely to separate two overlapping words, or if something else affected the position of nouns and content words. However, making an evaluation looking at related metrics to the ones which change could easily grow very heavy and time-consuming, and out of the scope of this project.

The technical evaluation could also have been improved through performing an examination of skewness and kurtosis across the different metrics, for finding the z-scores to see if any outliers heavily affected the data. This would have been a good thing to implement in order to validate our results.

Doing all of these improvements would not have been reasonable in this project, even doing one of them would not have been reasonable in the case of studying related metrics. But implementing one, especially the lastly mentioned one, could have further expanded the information to be inferred from the results.

## 6.3 Future work

Future studies within the area investigated in this project could fall within a broad scope depending on where the focus and interest lies. In this part a few suggestions for future studies are presented and discussed.

An obvious relevant future study to conduct is a study where a similar survey was presented to TextAD's target audience. Because of the limited scope of this study and limited access to individuals within that group it was not possible to implement. However as the results show a difference between perceived readability of a text and what happens with the linguistic metrics it would be interesting to see if the changes are beneficial to a certain group. Because our results point in two different directions it would be highly relevant to see which side of it people with reading disabilities end up on. If a trend is found it might help with future studies and applications to help them by furthering the understanding of what they struggle with or what is to be prioritized when summarizing for them. Furthermore, a comprehensive literature study on the target group in combination with tweaking the weighting of the indexes done in this project could lead to a better summarizer.

Further studies could be done for the weighting of the indexes. As mentioned in Section 3.7, we did some initial tests for finding the optimal weights. These tests were not fruitful, so we therefore ended up weighting each index equally. It would, however, be interesting to see whether there is an optimal set of weights, and whether some of the indexes are more important than others for readability. Studies could also be done on the indexes in isolation, as opposed to aggregating them. This would make apparent the effectiveness of each index.

In addition to modifying the weights, the rules for which sentences should be sorted can also be modified. This would be a more sophisticated use of the cohesion measures, as opposed to our approach of blindly maximizing the measures. For an example of this, during the development of CohSort, we found that excluding the first and last sentence from the sorting process, and thereby only sorting the sentences in between them, generally improved the perceived readability, at least among us. This, however, extended beyond the scope of the project, and was therefore not explored. It may therefore be a question for further studies.

Regarding the indexes, we relied only on sentence-to-sentence indexes, that is, they measure cohesion only between sentences. Cohesion can also be measured on higher levels, such as paragraphs. It would perhaps not be surprising to find that maximizing sentence-to-sentence measures, as we have done, sacrifices higher level cohesion. Thus future studies could focus both on evaluating how maximizing sentence-to-sentence cohesion indexes affects high-level cohesion measures, as well as what happens when maximizing these high-level measures.

Such a high-level measure could perhaps be the LSAPPI index in Coh-Metrix (McNamara et al., 2014, p. 66). It is similar to LSASS1 (see Section 2.5.1.1), but compares the cosine similarities between adjacent paragraphs rather than sentences. One could simply add this index to the aggregate in CohSort and test if this improves the readability similar to how we have done. Alternatively one could also develop a more sophisticated sorting method of the

sentences. For example, sentences within paragraphs could be sorted according to our CohSort approach, while these paragraphs are then sorted using LSAPP1. If this improves readability remains to be seen.

In this project we did also study newspaper articles, the same type of articles which ElsaSum was trained on (see Section 2.4). Applying CohSort to other types of texts may yield different results. Further, since ElsaSum is an extractive summarizer, our results may only apply to extractive summaries. CohSort may, for example, produce different results for abstractive summarizers.

## **6.4 Implications**

To conclude, the results from the two evaluations implicate two different conclusions, wherein the survey shows that the cohesion of the CohSort summary is lower than that of the ElsaSum summary. However the technical evaluation shows the opposite in terms of computational linguistic metrics.

As both the evaluations could have been improved and expanded a definite conclusion cannot be drawn as to which one should weigh heavier on a decision of using the LSA and L2 measurements in order to improve cohesion in text. Instead further testing should be done (see Section 6.3) in order to establish how well L2 and LSA can be used, but also in order to find an ideal evaluation for readability.

What has primarily been found through this study is that while the cohesion of text might improve according to computational measures, this is not a guarantee for it to be perceived as better or even equal to a similar text with lower scores. Readability is ultimately up to the reader. This therefore suggests that the cohesion measures do not fully capture readability as a phenomenon. This is not to say that cohesion measures are useless and that they do not capture readability at all. It merely suggests that there is more to readability than what the measures encompass, and that these limitations come to light when one tries to maximize the values of these measures as we have done in this study.

## **6.5 Ethics and sustainability in relation to text summarization**

Text summarization could potentially lead to more people being included in today's information driven society. This does however not come without challenges. One potential issue is that information can be altered or lost when a text is shortened through summarization. Access to free information is essential for a well functioning democracy, and providing an, in some cases, already marginalized group with less, or possibly incorrect, information could lead to an unlevel playing field. This highlights that the importance of keeping the text's message intact while making it accessible for all is a balancing act, and text adaptation applications must be developed with this in mind.

A further potential issue with text summarization, and text simplification in general, is that it may impede development of reading abilities. To become proficient in reading requires being

challenged by texts that one may find difficult. Text summarization tools may therefore remove this challenge for the reader, and thereby impede development. This is not to suggest individuals with reading difficulties should refrain from using these tools, or that these difficulties can simply be trained away.

An issue with artificial intelligence in general is its environmental impact. The processing power required to train and run these algorithms demands a large amount of resources, and the environmental impact of artificial intelligence can not be neglected. While the impact of this particular project is relatively low, it is an opportunity for us to explore ways to make the most of a finite amount of processing power by writing an as efficient algorithm as possible. By opting for less computationally expensive techniques, and not using more complex tools than needed for the application, we can save resources and limit the environmental impact of AI tools.

## 7. Conclusions

The purpose of this study was to investigate the role of automatic cohesion measures in regards to extractive text summarization, and ideally to find a more effective solution for TextAD. We sought to examine whether or not these measures can improve the readability of summaries by reordering their sentences. To conclude the study we return to our research question:

Can automatic cohesion measures improve the readability of summaries by reordering their sentences?

While our study does not provide an exhaustive answer, it still provides a partial answer. The results indicate that simply maximizing the sentence-to-sentence cohesion measures is not a viable approach to improve readability. We found that, while the summaries modified by CohSort fairly consistently rank higher than ElsaSum’s original summaries in computed linguistic metrics, readers do not actually find them more readable. This indicates a discrepancy between the computationally measurable linguistics and actual reader experience. While simply maximizing sentence-to-sentence cohesion is not a viable approach, it does, however, not leave out the possibility of increasing readability via more sophisticated uses of the cohesion measures.

# Bibliography

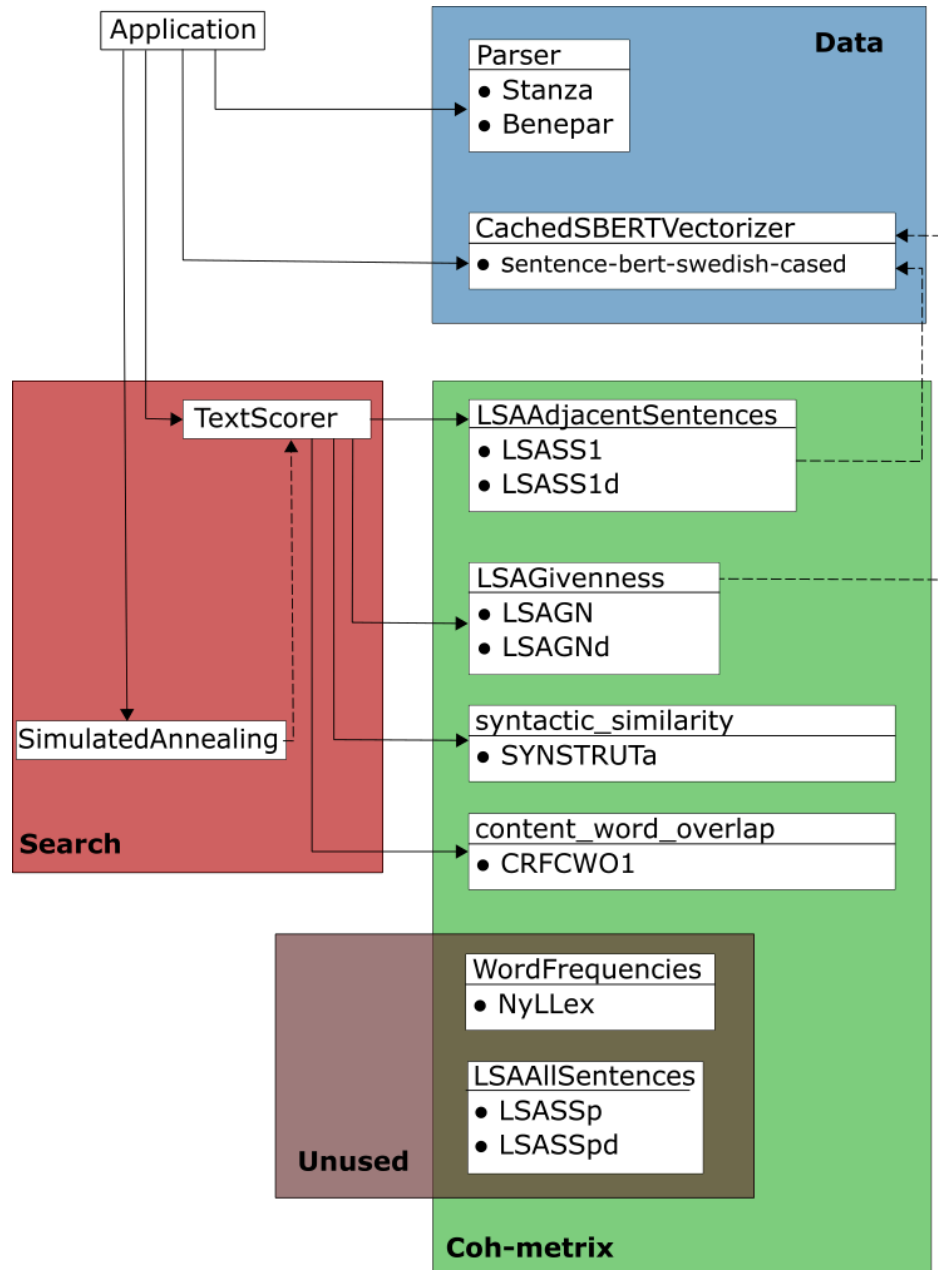
- Andersson, Elsa. 2022. *Methods for increasing cohesion in automatically extracted summaries of Swedish news articles: Using and extending multilingual sentence transformers in the data-processing stage of training BERT models for extractive text summarization*. B.A. Thesis, Linköping University.  
<https://www.diva-portal.org/smash/record.jsf?pid=diva2:1667268>
- Crossley, Scott A.; Greenfield, Jerry. and McNamara, Danielle S. 2008. Assessing Text Readability Using Cognitively Based Indices. *TESOL Quarterly* 42(3): 475-493.  
<http://www.jstor.org/stable/40264479>
- Dale, Edgar. and Chall, Jeanne S. 1949. The Concept of Readability. *Elementary English* 26(1): 19-26. <https://www.jstor.org/stable/41383594>
- Devlin, Jacob; Chang, Ming-Wei; Kenton, Lee. and Toutanova, Kristina. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*.  
<https://arxiv.org/abs/1810.04805> (Accessed 2023-05-20).
- El-Kassas, Wafaa S.; Salama, Cherif R.; Rafea, Ahmed A. and Mohamed, Hoda K. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications* 165(1): 113679. <https://doi.org/10.1016/j.eswa.2020.113679>
- Fahlborg, Daniel. and Rennes, Evelina. 2016. Introducing SAPIS – an API Service for Text Analysis and Simplification. In *The second national Swe-Clarin workshop: Research collaborations for the digital age*. Umeå, Sweden.  
<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-141550> (Accessed 2023-05-22).
- Gough, Philip B. and Tunmer, William E. 1986. Decoding, reading, and reading disability. *RASE: Remedial & Special Education* 7(1): 6-10.  
<https://doi.org/10.1177/074193258600700104>
- Graesser, Arthur C.; McNamara, Danielle S.; Louwerse, Max M. and Cai, Zhiqiang. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers* 36(2): 193-202. <https://doi.org/10.3758/BF03195564>
- Hagberg, Aric A.; Schult, Daniel A. and Swart, Pieter J. 2008. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*. Pasadena, CA USA, 11-15.
- Hahn, Udo. and Mani, Inderjeet. 2000. The challenges of automatic summarization. *Computer* 33(11): 29–36. <https://doi.org/10.1109/2.881692>
- Harris, Charles R.; Millman, K. Jarrod.; van der Walt, Stéfan J.; Gommers, Ralf; Virtanen, Pauli; Cournapeau, David; Wieser, Eric; Taylor, Julian; Berg, Sebastian; Smith, Nathaniel J.; Kern, Robert; Picus, Matti; Hoyer, Stephan; van Kerkwijk, Marten H.;

- Brett, Matthew; Haldane, Allan; Fernández del Río, Jaime; Wiebe, Mark; Peterson, Pearu; Gérard-Marchant, Pierre; Sheppard, Kevin; Reddy, Tyler; Weckesser, Warren; Abbasi, Hameer; Gohlke, Christoph. And Oliphant, Travis E. 2020. Array programming with NumPy. *Nature* 585(7825): 357-362.  
<https://doi.org/10.1038/s41586-020-2649-2>
- Holmer, Daniel. and Rennes, Evelina. 2022. NyLLex: A Novel Resource of Swedish Words Annotated with Reading Proficiency Level. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 1326-1331.  
<https://aclanthology.org/2022.lrec-1.141/> (Accessed 2023-05-24).
- Isaksson, David. 1970. *AI skapar texter för personer med lässvårigheter*.  
<https://liu.se/nyhet/ai-skapar-texter-for-personer-med-lassvarigheter>  
 (Accessed 2023-05-22).
- Janfalk, Ulf. 2019. *Linjär Algebra*. Matematiska institutionen, Linköping Universitet.
- Jurafsky, Daniel. and Martin, James H. 2023. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. <https://web.stanford.edu/~jurafsky/slp3/> (Accessed 2023-05-23).
- Kitaev, Nikita. 2018. *Berkeley Neural Parser*.  
<https://github.com/nikitakit/self-attentive-parser> (Accessed 2023-05-23).  
<https://spacy.io/universe/project/self-attentive-parser> (Accessed 2023-05-23).
- Landauer, Thomas K. 2013. LSA as a Theory of Meaning. In Landauer, Thomas K.; McNamara, Danielle S.; Dennis, Simon. and Kintsch, Walter (eds.). *Handbook of Latent Semantic Analysis*. Routledge.
- Linderholm, Tracy; Everson, Michelle G.; van den Broek, Paul; Mischinski, Maureen; Crittenden, Alex. and Samuels, Jay. 2000. Effects of Causal Text Revisions on More- and Less-Skilled Readers' Comprehension of Easy and Difficult Texts. *Cognition and Instruction* 18(4): 525-556. [https://doi.org/10.1207/S1532690XCI1804\\_4](https://doi.org/10.1207/S1532690XCI1804_4)
- MacLellan, Christopher. 2019. *Py Search*. [https://github.com/cmaclell/py\\_search](https://github.com/cmaclell/py_search) (Accessed 2023-05-24).
- Margarido, Paulo R. A.; Pardo, Thiago A. S.; Antonio, Gabriel M.; Fuentes, Vinícius B.; Aires, Rachel; Aluisio, Sandra M. and Fortes, Renata P. M. 2008. Automatic summarization for text simplification. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*. New York, USA: Association for Computing Machinery, 310-315. <https://doi.org/10.1145/1809980.1810057>
- McNamara, Danielle S.; Kintsch, Eileen; Songer, Nancy B. and Kintsch, Walter. 1996. Are good texts always better? Interactions of text coherence, background knowledge, and

- levels of understanding in learning from text. *Cognition and instruction* 14(1): 1-43. <https://www.jstor.org/stable/3233687>
- McNamara, Danielle S.; Graesser, Arthur C.; McCarthy, Philip M. and Cai, Zhiqiang. 2014. *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press.
- Menken, Caroline M. and Toepoel, Vera. 2023. How to Optimize Online Mixed-Device Surveys: The Effects of a Messenger Survey, Answer Scales, Devices and Personal Characteristics. *Methods, data, analyses : a journal for quantitative methods and survey methodology*. 17(1): 47-70. <https://doi.org/10.12758/mda.2022.08>
- Monsen, Julius and Jönsson, Arne. 2021. A method for building non-English corpora for abstractive text summarization. In *Proceedings of CLARIN Annual Conference*. Linköping University Electronic Press.
- Monsen, Julius. and Rennes, Eevelina. 2022. Perceived Text Quality and Readability in Extractive and Abstractive Summaries. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 305-312. <https://aclanthology.org/2022.lrec-1.32> (Accessed 2023-05-22).
- OpenAI. 2023. *ChatGPT-4*. <https://openai.com/product/gpt-4> (Accessed 2023-05-23).
- Orăsan, Constantin. 2019. Automatic summarisation: 25 Years on. *Natural Language Engineering* 25(06): 735-751. <https://doi.org/10.1017/S1351324919000524>
- Reimers, Nils. and Gurevych, Iryna. 2019. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. <https://arxiv.org/abs/1908.10084> (Accessed 2023-05-20).
- Rekathati, Faton. 2023. *The KBLab Blog: Swedish Sentence Transformer 2.0*. <https://kb-labb.github.io/posts/2023-01-16-sentence-transformer-20/> (Accessed 2023-05-24).
- Russel, Stuart J. and Norvig, Peter. 2016. *Artificial Intelligence: A Modern Approach*. Pearson Education Limited.
- Stanford NLP Group. 2020. *Stanza - A Python NLP Package for Many Human Languages*. <https://stanfordnlp.github.io/stanza/> (Accessed 2023-05-24).
- Universal Dependencies. 2014-2022. *Universal POS tags*. <https://universaldependencies.org/u/pos/> (Accessed 2023-05-24).
- University of Memphis. n.d. *Coh-Metrix version 3.0 indices*. [http://cohmetrix.memphis.edu/cohmetrixhome/documentation\\_indices.html#LSA](http://cohmetrix.memphis.edu/cohmetrixhome/documentation_indices.html#LSA) (Accessed 2023-05-24).

# Appendix A

The structure of the CohSort application.



## Appendix B

### Sammanfattning av informationssökning

#### Hitta sökord

Vår informationssökning utgick till stor del från TextAD-gruppens publikationer och rekommendationer. Vi har läst deras publikationer och genom dessa, och deras källor hittat originalkällorna för olika tekniker och algoritmer. För att hitta relevanta källor till övergripande teori om textsammanfattning utgick vi från sökord som “automatic text summarization” och “text summarization NLP”. Dessa var mycket användbara för att hitta allmän information om textsammanfattning och översiktsartiklar, men vi behövde snart gå vidare till källor som mer specifikt behandlade den typ av sammanfattning vi arbetar med. Då var sökord som “extractive summarization” och “extractive summarization transformers” mer användbara. När vi letat information om läsning och läsförståelse har vi följt samma mönster, och börjat generellt med breda söktermer som “reading” och “reading comprehension”. Utifrån de första träffarna kunde vi konstatera att läsförståelse kan delas upp i olika delar, och följaktligen blev dessa våra nya sökord: “reading decoding” och “linguistic comprehension”.

Såväl de breda som de mer smala sökorden har gjort nytta i projektet. Utan breda sökningar som “reading comprehension” eller “text summarization” hade vi inte kommit vidare och funnit våra smalare söktermer som “linguistic comprehension” eller “extractive summarization”. Det är alltså en kombination av sökord som varit mest användbara, dock har vi som nämnts ovan inte behövt göra informationssökning relaterat till själva projektet från grunden, utan många källor var givna från början och det var snarare en fråga om att navigera bland dessa och hitta vad som var viktigt än att söka från grunden. Exempelvis så har informationssökningen kring de mått och tekniker vi applicerat till största del kommit från publikationer med direkt koppling till dessa, som boken om Coh-Matrix som källa till teorin om Coh-Matrix-mått.

#### Använda sökverktyg

I vår informationssökning har vi använt oss av flera olika sökverktyg. Det första av dessa är unisearch, som varit användbart för att hitta vetenskapliga artiklar. Vi har även använt oss av DiVA, där vi enkelt kunnat hitta de publikationer som är kopplade till textAD. Eftersom språkteknologi på svenska är ett så pass smalt område, där textAD är en av få aktörer, har det varit självklart för oss att utgå från dem. Vi har även sökt efter artiklar i ACM digital library.

Unisearch är bibliotekets ämnesöverskridande sökverktyg. Det innehåller mer än 700 miljoner artiklar, böcker och tidskrifter som finns tillgängliga via LiU. Sökgränssnittet kräver lite tid att sätta sig in i om man jämför med vanliga sökmotorer som Google, och upplevs inte som särskilt bra. Vid något tillfälle har ett stavfel smugit sig in när vi sökt efter en specifik titel och då hittas den inte alls. I google hade detta inte varit något problem. Det är lätt att få

ett väldigt stort antal träffar och man behöver generellt avgränsa sin sökning. Detta kan göras på flera sätt, exempelvis genom att begränsa till vissa publikationsår eller specifika discipliner. Det går också att avgränsa sökning genom att formulera sökningen med booleska operatorer. Genom verktyg för avancerad sökning är det också möjligt att sätta automatiskt AND eller OR genom att välja "hitta alla mina söktermer" eller "hitta några av mina söktermer". Det går även att begränsa till peer-reviewed tidskrifter för att säkerställa kvalitén. Källan går att komma åt via sökverktyget, förutsatt att man valt att bara få träffar som är tillgängliga via LiU. Man får dock först klicka på fulltext och sen vidare till fulltexten på den sida som kommer upp.

DiVA (via biblioteket) innehåller publikationer av forskare och studenter vid svenska universitet och högskolor inom alla områden. Om man enbart söker på dokument med fulltext i DiVA får man 595 000 träffar. Dessa går bland annat att sortera på uppsatsnivå, organisation och person. Sökträffarna kan ordnas efter författare, publiceringsdatum eller titel. Det går att söka bland studentuppsatser och/eller forskningspublikationer. Sökningen är överlag väldigt lik den i unisearch, och man kan avgränsa på olika sätt, använda booleska operatorer och sortera sitt sökresultat på samma sätt. Man kan också välja att enbart få peer-reviewed träffar genom att välja refereed. Detta sorterar dock bort doktorsavhandlingar, som vi velat läsa, och vi har därför inte använt oss av den funktionen på samma sätt på DiVA. Det känns lite lättare att överblicka resultatet i DiVA jämfört med unisearch, men kanske beror det bara på själva gränssnittet som är lite mer minimalistiskt. Detta gör att DiVA upplevs som lite mer användarvänligt, men det tar ändå lite tid att lära sig som ny användare. Man kommer direkt till fulltexten från sökresultatet.

ACM Digital Library är en databas innehållandes över 700 000 fulltext-artiklar inom IT och programmering. Det är möjligt att söka inom ett område (som artificiell intelligens) eller i hela datasamlingen. Sökningen är onödigt krånglig, genom att ACM automatisk sätter OR mellan söktermerna. För att söka på extractive summarization behövde vi därför skriva antingen "extractive summarization" eller extractive AND summarization, för att hitta relevanta artiklar. Det går att skapa sökfrågor och använda söktekniker på liknande sätt som de andra verktygen, genom att exempelvis använda booleska operatorer eller sortera på senast publicerat. Träfflistan kan sorteras på relevans, datum, citeringar eller antal nedladdningar. Här skiljer ACM sig lite från de andra, som inte går att sortera på citeringar och antal nedladdningar. ACM innehåller också något de kallar "reproducibility badges", som exempelvis markerar om ett resultat kunnat replikeras eller reproducerats. Alla artiklar kommer från ACMs publikationer, som alla är peer-reviewed. Dessa är tillgängliga i fulltext via sökverktyget.

## Att söka efter information

En av våra första sökningar i unisearch gjordes med sökorden "text summarization". Detta gav ett stort antal träffar, som var svåra att överblicka. En senare men liknande sökning gjordes med sökorden "text summar\* AND (evaluation OR assessment)", när vi letade efter metoder för utvärdering av textsammanfattare. Att sätta wildcard (\*) innebär att vi får med

träffar med summarization, summary, summarisation och summaries. Vi vill också bara ha träffar som innehåller både textsummering och utvärdering, och använder därför AND. OR används för att leta efter antingen evaluation eller assessment, som båda är begrepp som vi stött på i vår informationssökning. Vi har dessutom avgränsat sökningen till peer reviewed-artiklar och till artiklar som är mindre än 5 år gamla. Genom att använda de söktekniker och funktioner vi lärt oss om under projektet blir det lättare att hitta relevant information och få svar på riktiga specifika frågor.

## Källkritik och urval

En av de källor vi använt oss av i rapporten är artikeln *Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding* av Devlin et al. Samtliga författare arbetar eller har arbetat vid Google AI language. Detta är den artikel där Google AI Language introducerade sin språkmodell BERT, som den summerare vi använder oss av bygger på. Den finns publicerad i *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. The Association for Computational Linguistics (ACL) är en organisation för forskare och yrkesverksamma inom naturligt språk-behandling. Det är också denna grupp, samt studenter inom området, som vi bedömer att källan riktar sig till. Artikelns källor finns tydligt angivna i referenslistan. Det är en mycket välkänd och citerad artikel inom området, som enligt google scholar hänvisats till över 66000 gånger. Källan är från 2019, som inom IT-området kan tyckas något föråldrat. Särskilt inom ett område som artificiell intelligens, där nya framsteg görs hela tiden. Vi bedömer dock att källan fortfarande är aktuell för det vi använder den till, det vill säga information om språkmodellen BERT. Medan den med all säkerhet utvecklats sedan artikeln skrevs är grundprinciperna och tekniken densamma, och det är denna vi hämtat information om från artikeln. Man kan också tänka sig att källan inte är neutral när den diskuterar bakgrunden, implikationer och andra konkurrerande språkmodeller eftersom Google är en av de största aktörerna på området. Den information vi hämtat från artikeln rör dock enbart deras språkmodell och tekniken bakom, som de måste sägas vara den mest trovärdiga källan för. I samband med att artikeln publicerades på konferensen verkar den, utifrån vad vi kan läsa om konferensen riktlinjer, ha blivit peer-reviewed. Vi har inte hittat någon källa som motsäger det som står i artikeln, men vi kan tänka oss att viss information inte är aktuell längre eftersom området snabbt utvecklas. Detta är också något som varit en utmaning för projektet generellt. Det är en balansgång mellan att använda äldre förstahandskällor och nya uppdaterade källor. Så länge man är noga med vilken typ av information som hämtas från olika typer av källor fungerar det dock bra. Ett annat gott tecken är att många av de äldre källorna återkommer citerade i nyare publikationer. Det kanske enklaste har varit att källkritiskt bedöma de artiklar som handlar om de tekniker och mått vi implementerat i programmeringsdelen av projektet. Dels finns det generellt inga alternativa källor än de som getts ut av de som uppfunnit teknikerna, och dels kan vi direkt testa dem i programmeringsmiljön och se hur de fungerar på riktigt. Den typen av information är inte heller något som färgas av åsikter på samma sätt som exempelvis psykologi, och kan generellt beskrivas rent matematiskt.

## Utvärdera informationssökningen

Den information vi hittat under vårt arbete med projektet har bidragit till att besvara frågeställningen genom att ge oss djupare kunskap om de verktyg vi använt och förståelse för hur dessa kan implementeras. Vi har också fått en djupare förståelse för textsammanfattning och läsning som helhet tack vare informationssökningen. Det som varit både en utmaning och en tillgång är att det många gånger bara funnits en eller ett fåtal källor om det aktuella ämnet. Det har gjort själva sökningen enkel, men utvärderingen svårare eftersom det inte funnits något att jämföra med. Det svåraste har överlag varit att läsa sig in på ämnet textanpassning, som vi var bekanta med men inte på så teknisk nivå, för att veta var vi skulle börja söka. Det var också lite av en utmaning att lära sig söka på ett effektivt sätt med exempelvis booleska operatorer och veta hur man ska sortera resultaten. Det här är dock något vi kommer att kunna använda oss av nästa gång vi gör något liknande, och som därför inte kommer kräva lika mycket tid i framtiden. Det vi har lärt oss är vikten av att börja med en bred sökning och informationsinhämtning, och sedan avgränsa det till mindre områden. Genom att använda andra publikationer (från textAD-gruppen) och kolla på deras källor, och de källor de i sin tur använt, har vi kunnat få en överblick av området och identifierat relevanta artiklar. Detta har också varit ett bra sätt att hitta förstahandsinformation som fortfarande är aktuell. Vi har blivit bättre på att snabbt avgöra om en artikel är relevant eller inte genom att läsa den på det sätt som föreslagits under kursen (abstract, diskussion, introduktion, resultat, metod) och det är något som kommer att bli mycket användbart i kommande kurser och våra kandidatuppsatser. Vad gäller källvärdering har vi blivit bättre på att välja *vad* vi hämtar från olika källor. Som vi nämnade ovan så hämtar vi inte information om språkmodeller generellt från Googles artikel, utan enbart om deras språkmodell BERT. På så sätt kan en källa som inte är neutral av exempelvis ekonomiska skäl ändå användas.

# Appendix C

## Summaries

### Summary 1

Natten till den 13 maj kan den som gillar att lyssna på fågelsång, men är för trött för att själv sitta på en stubbe i skogen och vaka när vårfåglarna vaknar, knäppa på radions P1. Möjligen kan man betrakta mitten på maj som startskottet för nattsångarna. Men var på plats redan vid midnatt för att fånga in de riktiga nattsångarna. Så fort gryningsljuset kommer, blandas deras nattliga sång upp med morgonserenader från alla håll av rödstjärtar, trastar, bofinkar, rödhakar. Var söker man de nattaktiva fåglarna, utöver ugglornas hoande ?. Där finns gott hopp om att höra surrande gräshoppsångare eller ute i vassen kärresångare, rörsångare och trastsångare. En lite udda nattaktiv fågel är nattskärra, som får sökas i helt annan miljö. Man ska ha tur om man får syn på en väl kamouflerad nattskärra, som brukar jämföras med en brunmelerad barkbit där den ligger på stigen eller längs en grov trädgren.

### Summary 2

I sitt regleringsbrev för 2022 fick Myndigheten för kulturanalys i uppdrag av den förra regeringen att utreda hur deltagandet i kulturlivet skulle kunna breddas och hur snedrekryteringen till kultursektorn skulle kunna minskas. I den rapport som nu publicerats konstaterar man att den främsta anledningen till varför människor med lågt kulturdeltagande väljer bort kulturaktiviteter är ointresse. Även tidsbrist, kostnader och avsaknad av intressant kulturutbud nära hemmet spelar in. Samtidigt fortsätter faktorer som utbildningsnivå, inkomst, kön, ålder och utländsk bakgrund att vara avgörande för skillnader i hur befolkningen tar del av kultur. Myndigheten slår nu fast att två kulturpolitiska åtgärder är särskilt viktiga för att skillnaderna i deltagande ska minska. Det handlar dels om att prioritera åtgärder för att nå alla barn och unga, dels om att förstärka den nationella infrastrukturen för kultur, till exempel genom stöd till kommuner och organisationer i civilsamhället. När det gäller barn och unga bedöms ett förstärkt stöd till kommunala verksamheter, exempelvis ett fortsatt nationellt stöd till kulturskolan, långsiktigt kunna bidra till att minska de skillnader som nu finns i förutsättningarna för deltagande ", säger hon i ett pressmeddelande ". Men arbetet för att skapa likvärdiga möjligheter för deltagande och professionellt arbete i kultursektorn kräver framför allt åtgärder inom andra områden, exempelvis inom utbildningspolitiken och arbetsmarknadspolitiken ".

### Summary 3

Sverige ska vara bäst i världen på att använda digitala verktyg i skolan. I juni förra året fick Skolverket uppdraget att uppdatera den nationella digitaliseringsstrategin som ska gälla för skolan de närmaste fem åren. Det är en skrivning som är otydlig eftersom det är oklart vad digital kompetens faktiskt innebär för barn i förskoleåldern. Det är inte ens säkert att barn under sex år behöver tillägna sig digital kompetens, menar IFAU. Institutet lyfter fram att det saknas forskning om hur digitala verktyg

påverkar kunskapsutvecklingen på förskolan och i lågstadiet. Därför är det viktigt att Skolverket är mer försiktig i skrivningarna som gäller de yngre barnen, menar IFAU. Den digitala strategin kan lätt tolkas som att digitala verktyg ska införas tidigt. Skolverket får det att låta som att undervisningen och likvärdigheten blir bättre i takt med en ökad digitalisering, menar IFAU.

#### **Summary 4**

Jämfört med för tre månader sedan har genomsnittspriset på en bostadsrätt i Sverige stigit med 4 procent samtidigt som villapriset stigit med 1 procent. Men jämfört med för ett år sedan är det fortfarande minussiffror som gäller. Jämfört med mars har bostadsrättspriset stigit med 2 procent i Storgöteborg, Stormalmö och centrala Malmö. Genomsnittspriset på villor i april steg med 1 procent i Storstockholm och Stormalmö, jämfört med mars. Jämfört med för ett år sedan har villapriset fallit med 15 procent i Storstockholm, och Stormalmö och med 16 procent i Storgöteborg. Det ska understrykas att marknaden fortfarande är trög. Flera mäklare som kommenterar statistiken spår att Sverige snart nått toppen på Riksbankens räntehöjningar, den senaste i slutet av april. Genomsnittspriset på en bostadsrätt i Sverige har stigit med 4 procent på tre månader.

#### **Summary 5**

I veckan väntas temperaturen stiga flera grader, både natt och dag, i hela landet. Högtrycket som legat över Sverige de senaste dagarna, tillsammans med sval luft, byts då ut mot varmare luft och lågtryck -. Under måndagsdygnet slår det om till ostadigare väder, med fronter och nederbörd. Samtidigt lyckas högtrycket hålla lågtrycket borta ett tag till i söder så där finns utrymme för fortsatt soligt väder under måndags- och tisdagsdygnet, säger Erik Højgård - Olsen som är meteorolog på SMHI. I söder kan dagstemperaturen ligga på drygt 15 grader den kommande veckan, med nästan lika varmt väder längst Norrlandskusten och omkring 10 grader i Norrlands inland. Våren, och framför allt början av maj, har varit flera grader kallare än normalt hittills. Och våren som helhet är ungefär likadan, fast med lite mindre skillnader. Erik Højgård - Olsen säger att en kall vår inte säger något om hur sommarvädret blir.

#### **Summary 6**

I den första semifinalen vann skåningarna enkelt med 26 - 20 i Kristianstad arena. Att hemmaplan är oerhört betydelsefull i handboll gavs ett bevis på när Hammarby vann stort i returmatchen i Stockholm. Men efter 23 minuter var det färdigspelat för viktige Martin Dolk. Nu gäller det för Hammarby att knipa en match i Skåne. Kristianstad har hemmaplanfördel i en eventuell femte och avgörande match. Historiskt har Hammarby haft det jobbigt mot just Kristianstad. Hammarby visade dock i den första finalen i Svenska cupen tidigare i vår att laget kan vinna i Kristianstad ( 31 - 30 ). Hammarbys Viktor Ahlstrand jublar efter ett mål i semifinalen mot Kristianstad.

## **Summary 7**

2 - 0 i premiären mot England följdes upp med 2 - 0 även i den andra gruppsspelsmatchen, mot Kina. Anna Nordqvist och Caroline Hedwall kopplade tidigt greppet i sin bästboll mot Liu Yu och Liu Ruixin. Det är därför det är så himla kul att spela med "Carro", för man vet att hon älskar press och presterar bland den bästa golfen man ser under press, säger Anna Nordqvist i en intervju med LPGA - touren. Madelene Sagström och Maja Stark avgjorde också på det 17:e hålet i sin match mot Yin Ruoning och Lin Xiyu. Det känns som att vi har haft väldigt kul den här veckan och som att det har varit till vår fördel, säger Anna Nordqvist inför fortsättningen. Semifinalerna och finalen på söndag avgörs på ett litet annat sätt. Fakta. International Crown Anna Nordqvist gör succé i International Crown tillsammans med Caroline Hedwall, Madelene Sagström och Maja Stark.

## **Summary 8**

Museet är underjordiskt och ligger under Norrbro på Helgeandsholmen. Där har museet legat ända sedan 1986, men den 5 november i år behöver de flytta ut. Enligt beslut i Kulturnämnden ska Medeltidsmuseet i stället flytta in Börshuset i Gamla stan, där Nobelmuseet just nu huserar. Nobelmuseet ska i sin tur flytta till det ännu obbyggda Nobel Center vid Stadsgårdskajen när Slussenbygget är klart -. De kan vi så klart inte ta med oss, men det är en fullgod utställningslokal som vi kan bygga en jättefin utställning om medeltiden i, säger Lin Annerbäck, museichef på Medeltidsmuseet. Men fram till dess väntar alltså flera år utan lokal för Medeltidsmuseet -. Beskedet att Medeltidsmuseet behöver flytta har mött motstånd, inte minst bland insändare i DN -. Lin Annerbäck lyfter fram fyndet av Stockholms stadsmur som Medeltidsmuseet är uppbyggt kring -.

## **Summary 9**

Granskningen av litteraturprofilen har varit på gång under en längre tid och har på förhand varit omtalad i flera medier, däribland Expressen och Tidningen Vi. Men nu väljer "Uppdrag granskning" att pausa den planerade publiceringen, något som Aftonbladet var först med att rapportera -. Det är svårt att ge detaljer om en opublicerad granskning, men i det här fallet handlar det om att vi måste känna oss trygga i att vi kan erbjuda tillräckligt skydd till de som medverkar, och det har vi inte gjort, säger Axel Björklund, ansvarig utgivare för "Uppdrag granskning". Detsamma gäller när granskningen var tänkt att publiceras. Enligt uppgifter till DN skulle den ha sänts den 10 maj. Innan beslutet om att pausa publiceringen blev offentligt fick SVT ta del av en utredning som advokat Johan Eriksson på advokatbyrån Försvarsadvokaterna gjort på uppdrag av det företag som litteraturprofilen är kopplad till. Dokumentet, som DN tagit del av, består av 75 punkter och drar bland annat slutsatsen att en sändning skulle kunna innebära en "oförsvärlig publicitetsskada". Enligt Axel Björklund finns ingen reell jävsituation att tala om -.

## **Summary 10**

I kommunägda Business region Göteborgs rapport, som publicerades på tisdagen, anges det lokala konjunkturindexet till 94,2. Men tjänsteföretagen, där många är

beroende av industrin, är försiktigt positiva, säger Henrik Einarsson, analyschef på Business region Göteborg. Som väntat är siffrorna dystrare för handel och bygg ". Utsikterna framåt inom regionens byggsektor är fortsatt mycket pessimistiska vad gäller byggvolym och anställda ", skriver BRG. Redan i förra kvartalsrapporten befann sig Göteborgsregionens handelssektor i en lågkonjunktur, enligt index. Även om Business region Göteborg tror att SCB kan ha överskattat utvecklingen en aning - plus 5,3 procent - beskrivs den som " klart starkare " än Stockholmsregionens ( +1,1 procent på årsbasis ) och Malmöregionens ( +4,4 procent ). Arbetslösheten var i april 5,3 procent i regionen, klart lägre än snittet under åren 2014 - 2019, alltså före pandemin. Det är den högsta vakansgraden i Göteborg sedan 2006.

### **Summary 11**

Seko skriver att uppgörelsen inte löser branschens problem - som handlar om underbemanning efter att många tåganställda slutat - men att den är ett steg framåt. Enligt Seko hade arbetsgivarsidan ställt krav på att man skulle arbeta tre av fyra helger, i stället för varannan, något som man lyckats stoppa. Frågan om ensamarbete vid framförande av resandetåg och vid olycksdrabbade arbetsuppgifter ska hanteras i en partsgemensam arbetsgrupp. Även arbetsgivarsidan är nöjd med uppgörelsen och att lösningen på de sena schemaändringarna " som främst drabbar medarbetarna men inte heller är önskvärda för arbetsgivarna " , skriver Almega Tågforetagen i en kommentar ". Kravet på stopp för ensamarbete på vissa tåg - något som ett stort antal lokförare försökte driva igenom med en vild strejk tidigare i vår - har man alltså inte kommit längre med än att den ska utredas genom att en arbetsgrupp tillsätts. Efter att förhandlingar hela natten mellan fackförbundet Seko och arbetsgivarnas organisation Almega Tågforetagen presenterades uppgörelsen vid 10.00 - tiden på måndagen. Då avväjdes samtidigt den strejk i tre steg som Seko varslat om till klockan 15.00 på måndagen. Den skulle i första hand ha drabbat tågtrafiken i Skåne och Öresundstågen.

### **Summary 12**

Den 7 juli förra året sköts en ung man till döds på Länsmanstorget på Hisingen i Göteborg. Polisen placerade mordet i den blodiga gängkonflikt som har pågått i Biskopsgården i mer än tio år. Fyra personer från gänget i södra Biskopsgården dömdes av tingsrätten för mord eller medhjälp till mord. Att de unga männen släpps ur häkte indikerar tydligt att de kan komma att frias från brottsmisstankarna. I fredags greps en person på Hisingen misstänkt för grovt vapenbrott. Polisen gick ut med ett pressmeddelande där Daniel Norlander, lokalpolisområdeschef på Hisingen, sade sig vara säker på att man hade förhindrat ett mord. Polisen undersöker om det var de nyligen frigivna männen som var måltavlor för vad de tror var ett planerat mord -. Läget i Biskopsgården är upphettat och polisen höjde i fredags beredskapen för våldshandlingar till rött läge -.

### **Summary 13**

Donald Trump hävdar att de politiska motståndarna var i maskopi med den federala polisen FBI för att sätta dit honom. Han hämtar stöd från en ny rapport från den

särskilda åklagaren John Durham, som har granskat FBI : s utredning om kopplingarna mellan Ryssland och Trumpkampanjen 2016. Det ser inte ut som om någon kommer att fällas för det som Donald Trump kallat för " århundradets brott ". Däremot blir kritiken från Durham nytt bränsle i Donald Trumps attacker på åklagare och domare. Han har all anledning att försöka undergräva förtroendet för rättsväsendet. Vi har vant oss vid att demokratin är under press i USA, men presidentvalet 2024 kommer också att handla om hoten mot rättsstaten, i ett land där juridik och politik riskerar att smälta samman. Just nu tyder mycket på att valkampen blir en repris från 2020, med Trump och Biden i huvudrollerna. Men i grunden är det valet 2016 som fortsätter att kasta en skugga över amerikansk samhällsdebatt.

#### **Summary 14**

För över 80 år sedan övergav Thailand absolut monarki och lämnade över största delen av makten till parlamentet. Men vägen mot demokrati har varit krokig. Inför årets val har DN pratat med flera personer på landsbygden som berättar om partirepresentanter som erbjuder dem pengar mot en röst. Thailands demokrati är alltså långt ifrån perfekt. Ändå finns en viss optimism bland dem som stöder oppositionspartier. Men de måste vinna en jordskredsseger för att ha en chans att ta över styret. Flera thailändare DN träffar är skeptiska till att valet blir rättvist. En vallokal förbereds i staden Narathiwat, inför söndagens val i Thailand.

#### **Summary 15**

I opinionsundersökningarna ligger hans stöd på strax över 49 procent, vilket innebär att Erdogan går in i valet i underläge för första gången på många år. Två val hålls på söndagen - parlamentsval och presidentval. Sent på söndagskvällen väntas de första siffrorna från rösträkningen. Valdeltagandet ser ut att bli det högsta på länge, med över fem miljoner förstagångsväljare. Dessa röster tar längre tid att räkna än dem som avlagts på valdagen i Turkiet. Flera saker ger trots allt Erdogan ett visst övertag i slutspurten, inte minst kontrollen över medierna. Om ingen av kandidaterna får över 50 procent av rösterna på söndagen hålls en andra valomgång om två veckor. Anhängare till Kemal Kilicdaroglu deltar i ett valmöte i Ankara på fredagen.

## Appendix D

All of the keys changed over the comparisons of the summaries from all 15 articles sent to SAPIS. The keys are directly taken from the dictionaries returned from SAPIS for every request. The first column contains the key, and the second a count for how many of the comparisons had the key changed. The metrics analyzed in this study are italicized, within the report their names have been modified in order to make them easier to read and understand. However the order is the same as in the report and the names used in the report include the key words in the key names.

Changed key	Count
coh-metrixLSAadjacent_lsaavgavg	12
coh-metrixcohesionadjacentcoh-metrixcontent wordstdstd	12
coh-metrixcohesionglobalcoh-metrixanaphorsanaphors	12
screamssurfacesentence_lengthssentence_lengths	12
coh-metrixLSAadjacent_lsastdstd	12
coh-metrixcohesionadjacentcoh-metrixcontent wordsratio	12
coh-metrixcohesionadjacentcoh-metrixstemsstems	11
coh-metrixcohesionadjacentcoh-metrixargumentsargument s	11
coh-metrixcohesionadjacentcoh-metrixnounsnouns	10
coh-metrixcohesionglobalcoh-metrixcontent wordsratio	8
coh-metrixcohesionadjacentcoh-metrixanaphorsanaphors	5
screamstructuraln_nominal_postmodifiersn_nominal_post modifiers	4
screamstructuralavg_nominal_postmodifiersavg_nominal_ postmodifiers	4
screamstructuraldep_type_dictETET	4
screamstructuraldep_unigram_probsETET	4
screamssurfacesentence_length_sdsentence_length_sd	4
screamstructuralright_dependency_ratio	3
screamstructuralavg_dep_distance_dependentavg_dep_dis tance_dependent	3
screamstructuralavg_dep_distance_sentenceavg_dep_dista nce_sentence	3
screamstructuraldep_type_dictOAOA	3

screamstructuraldep_unigram_probsOAOA	3
screamstructuralverb_arity_unigram_probs11	3
screamstructuraln_right_dependenciesn_right_dependencies	3
screamstructuralverb_arities_dict11	3
screamstructuraltotal_dep_distancetotal_dep_distance	3
screamstructuralverbial_root_ratio	3
screamstructuraln_verbal_rootsn_verbal_roots	3
screamstructuralavg_verbal_arityavg_verbal_arity	3
coh-metrixcohesionglobalcoh-metrixcontent wordsstdstd	3
screamstructuraltotal_verb_aritytotal_verb_arity	3
coh-metrixcohesionglobalcoh-metrixstemsstems	2
screamstructuralavg_sentence_depthavg_sentence_depth	2
screamstructuraldep_type_dictDTDT	2
screamstructuraldep_type_dictOOOO	2
screamstructuraldep_unigram_probsOOOO	2
screamstructuraldep_unigram_probsHDHD	2
screamstructuraldep_type_dictHDHD	2
screamstructuralverb_arity_unigram_probs55	2
screamstructuraldep_unigram_probsDTDT	2
screamstructuraldep_unigram_probsVGVG	2
screamstructuraltotal_sentence_depthtotal_sentence_depth	2
screamstructuralverb_arities_dict55	2
screamstructuraldep_unigram_probsSSSS	2
screamstructuraldep_type_dictVGVG	2
screamstructuraldep_type_dictSSSS	2
screamstructuraldep_unigram_probsANAN	1
screamstructuralpos_tag_dictABAB	1
screamstructuralnominal_rationominal_ratio	1
screamstructuraldep_unigram_probsTATA	1
screamstructuraldep_type_dictANAN	1
screamstructuraldep_unigram_probsAAAA	1
screamstructuraldep_unigram_probs+F+F	1
screamstructuralpos_tag_dictJJJJ	1
screamstructuraldep_type_dictAAAA	1
screamstructuraldep_type_dict+F+F	1

screamstructuralpos_unigram_probsABAB	1
screamstructuraldep_type_dictTATA	1
screamstructuralverb_arities_dict33	1
screamstructuralverb_arities_dict44	1
screamstructuralverb_arity_unigram_probs33	1
screamstructuralverb_arity_unigram_probs44	1
coh-metrixcohesionglobalcoh-metrixnouns nouns	1
screamsurfaceavg_word_lengthavg_word_length	1
screamfacetotal_word_lengthtotal_word_length	1
screamstructuraldep_type_dictESES	1
screamstructuralpos_unigram_probsJJJ	1
coh-metrixcohesionglobalcoh-metrixarguments arguments	1
screamstructuraldep_unigram_probsESES	1

## **Appendix E**

The survey used in the study. It follows on the next page.