Visual Summarization of Meeting Transcripts

Master's Thesis

Silas Rudolf

rudolsil@students.zhaw.ch

Centre for Artificial Intelligence (CAI)
Natural Language Processing
ZHAW

Supervisor:
Prof. Dr. Mark Cieliebak

February 23, 2023

Abstract

The exponential growth in the amount of textual data generated daily makes it difficult to keep track of relevant information in everyday discussions, meetings, and other interactions. While in recent years, Natural language Processing NLP has been applied to several areas for extracting essential information from large volumes of text, only a limited amount of research was done in visualizing this information.

This thesis shows the incorporation of various NLP techniques, such as named entity recognition, sentiment analysis, and summarization, with a self-developed algorithm for keyword extraction and focuses on the visual presentation of this information to understand the critical information in a time-efficient and interactive way. The application is evaluated through a user test on a meeting transcript, comparing the response time of users utilizing the proposed application with those using the transcript only. The results show that with the help of the supporting application, the response time decreases on average by 15 seconds and up to over one minute on individual questions.

Contents

Abstract **i**

1 Introduction **1**

1.1 Motivation 1

1.2 Application Scenario 3

1.3 Goals 4

1.4 Challenges 5

1.5 Outline 5

2 Background **6**

2.1 Related Work 6

2.1.1 Text summarization 6

2.1.2 Dialogue summarization 8

2.1.3 Document visualization 11

2.2 Preliminaries 17

2.2.1 Semantic analysis 17

2.2.2 Architectures 23

2.2.3 Performance metrics 29

3 Design **32**

3.1 Problem definition 32

3.2 Design process 33

3.2.1 First iteration 33

3.2.2 Second interation 35

3.3 Final design and implementation 38

3.3.1 User interactions 39

4 Models and algorithms **41**
4.1 Entities . . . 41
4.2 Sentiment and Speaker Change . . . 43
4.3 Speaker Network . . . 45
4.4 Keywords and Summary sentences . . . 45
4.4.1 Summary sentences . . . 45
4.4.2 Keywords . . . 48

5 User test **52**
5.1 Setup . . . 52
5.2 Task design . . . 52
5.3 Evaluation . . . 54
5.3.1 Correctness of the responses . . . 55
5.3.2 Response times . . . 57
5.3.3 User feedback . . . 59

6 Conclusion **61**
6.0.1 Results . . . 61
6.0.2 Future Work . . . 62

Bibliography **63**

A Models and datasets **A-1**
A.1 Named Entity Recognition . . . A-1
A.2 Sentiment analysis . . . A-2
A.3 Summary sentences . . . A-2
A.4 ASR evaluation . . . A-3

B User Test results **B-1**
B.0.1 Questions . . . B-1
B.0.2 Application users . . . B-1
B.0.3 Transcript users . . . B-7

Chapter 1

Introduction

Many daily discussions or conversations today are available in some form of textual way. These discussions include transcriptions of important meetings and interviews, email threads, and social media interactions. In recent years, the quantity of those interactions has grown quite significantly. For example, some of the estimations for one single day is at around 500 million tweets [Livestats, 2022], 100 billion WhatsApp messages [Singh, 2020], 306.4 billion emails [Statista, 2022] and more than 11 million meetings[Visix, 2022].

With these large amounts of texts, it is often hard to recollect the information exchanged or know what was discussed as a non-participant in the discussion. Usually, more details are provided than needed, and filtering out essential information involves, in most cases, a tedious, time-consuming reading of the whole text.

This thesis focuses on improving this time-consuming process by applying specific Nature Language Processing architectures to compress and extract the essential information and visualization techniques to present and interact with the data.

1.1 Motivation

When we are presented with text in our daily life, how often do we read the text in its entirety? In many cases, only a short skimming of the text is required to get the essential information. The technology of text summarization can help with this knowledge extraction, allowing readers to view many documents for important information quickly.

Automatic text summarization produces a concise summary while preserving key information content and overall meaning. In recent years this has been applied widely in various domains. Examples are search engines that generate snippets as previews of the documents [Turpin et al., 2007] and news websites that produce condensed descriptions of news topics, usually as headlines, to facilitate browsing or knowledge extractive approaches [Allahyari et al., 2017]. Automatic

text summarization is very challenging, even for people. This difficulty comes from summarization requiring multiple complex natural-language-understanding components that involve information selection, evaluation, aggregation, and reorganization, followed by compression, generalization, or paraphrasing. Further, this must occur on multiple (possibly abstract) levels, such as sentences, paragraphs, sections, and documents. [Nikolov, 2020] When humans summarize a text, we usually read it in its entirety to get familiarized with its content and then write a summary highlighting its main points. However, since computers lack human knowledge and language capability, automatic text summarization is challenging and non-trivial.

In practice, there are two main approaches to text summarization. **Extractive summarization** extracts and combines text fragments precisely as they appear in the original document. Thus, the output is a compressed and reordered version of the input, with the original wording usually remaining the same. In **abstractive summarization**, the output is typically generated using a language model to produce novel sentences from information extracted from the corpus. Thus, summaries may contain new phrases and sentences that may not appear in the source text. [Nenkova et al., 2011]

In addition to those two distinguishing differences, the summary's output size can vary (e.g., a complete paragraph vs. a single sentence, a bullet point, several keywords, or even a single word). In return, this can significantly influence the transferred information and the reading time. Even though modern summarization systems are compelling, they are still far from reaching human-like performance and fluency. The output from the automated analysis is often too complex for data analysts to consume. The sense and decision-making based on the topic results rely on end-users, with the tasks often being exploratory and iterative.

Different analysis tasks or topics might require a different "level of depth" in the summarization (e.g., describing an overall topic of a discussion requires less information than answering a specific question about something mentioned during the conversation). Text analysis methods should be integrated with interactive visualizations to address this requirement, empowering the user by self-determining the summary's length/depth and context based on the information needed.

The hypothesis is that by presenting key ideas and abstract aspects of the text visually, enough information can be extracted to solve analytical tasks (such as answering questions about the discussion) that would otherwise require a full-text read-through. Combining visualization techniques and automatic algorithms should enable an effective and efficient union between the user and the machine-generated information. However, it is essential to note that this process is not a one-way road but an iterative process with feedback loops between different pipeline steps. The human input triggers and guides the steps of the automatic

analysis, which result in different representations of the data based on this input.

1.2 Application Scenario

A wide range of summarization variants is used in practical applications beyond the scope to be discussed. Instead, this document focuses on the following application scenario, which will influence the design and implementation of the text summarization models as well as the visualization and evaluation methods:

- The format of the analyzed input is in a multi-participant discussion, focusing mainly on business meetings, where time and speaker information are present.
- Content of the discussions is heterogeneous, has various topics, and is in a single language.
- Depending on the user, different topics are of interest, which makes the ability to interact dynamically with the data of need.
- Solely text data out of the discussion is used. Preliminary steps involving the transcription of audio files are not evaluated.
- Information used is static and has no live streaming data. The summarization and visualization are done as post-analysis.

In addition, discussions are defined utilizing the following characteristics:

- They are interactions between two or more persons.
- The key information of one dialogue is often scattered and spanned over multiple utterances and turns from different participants (which leads to low information density)
- The length of individual speaker utterances is not time-bound and can vary, thus leading to short, fragmented speech when the heat of discussion is high.
- The formality of the discussion depends on its setting and the connection of its speakers.
  - The degree of formality decides the discussion's level, structure, and rationality.
  - Speaker, topic (topic drifts), and timing can evolve and change spontaneously.

1.3 Goals

The concept presented in this thesis studies the application of text summarization methods combined with a visual presentation layer. The main goal of this approach is to **understand essential information of discussions in the most efficient way regarding time**. With this in mind, the following subgoals are considered, which allow a given user to:

- Assess the main topics of the discussion in less than 10 seconds.
- Find out what was discussed and get more context about any particular topic.
- Find out what a particular user said.
- View the general mood/sentiment of the discussion.
- Control the presented text according to the particular topic of interest.

The steps that are taken to achieve these goals are:

- Prototype and implementation of a parameterized summarization pipeline that can return the information with different levels of depth (e.g., paragraph, sentence, n-gram, keyword). The philosophy behind this is that it is upon the user to decide how much context is needed for a given task and that the infrastructure should be able to support a dynamic and interactive experience.
- Design of a visual presentation layer that can incorporate human interaction as an iterative process based on tests and user feedback.
- Implementation of the visualization on top of the summarization pipeline and evaluation of benefits and drawbacks. The evaluation of this approach is two-sided. One side is scoring the summary by standard evaluation metrics (ROUGE / BLEU); the other is defining a text analysis task that multiple subjects will perform.
- Gather and discuss insights into the NLP summarization process from a user perspective by knowing what information has to be displayed, what is essential, and how much information is needed to perform analysis tasks on an unseen/unknown discussion topic.

1.4 Challenges

Significant research efforts have been focused on summarizing single-speaker documents such as text documents, news, or scientific work, automatically summarizing and extracting essential information.

However, dialogue summarization, where multi-document analysis is needed, has received little attention despite the prevalence of dialogues and the vast application potential of dialogue summarization systems (meeting summary generation, customer service, media monitoring, and newsletters).

Furthermore, since dialogue language is inherently different from written text, it poses a unique set of challenges, making the development of automated methods to summarize and visualize this information more complex. Unlike single-document analysis, multiple-input documents are likely to contain more contradictory, redundant, and complementary information, and their relationships must be considered. [Ma et al., 2020]

Next to the summarization challenges is the visual aspect of text analysis, which is still a young field, and many topics still need to be explored. Among these are:

- Handling varying amounts of data. For example, meetings with a duration of 5 minutes vs. a 2-hour meeting.
- Displaying information across multiple dimensions, including time, current speaker, sentiment, and topic relevance.
- Control of granularity and transition between summarization levels and context.

1.5 Outline

The rest of this document is structured in the following way:

Chapter 2 presents background knowledge about automatic summarization and text visualization methods, together with essential research and related work in this field (focusing on the different modeling aspects of the problem and the visualization/user perspective part). Chapter 3 explains the design process of the visualization layer with its underlying models and algorithms described in chapter 4. An evaluation based on a practical experiment in the form of a user test is presented in chapter 5. Finally, the document concludes with a summary of the main findings and an outlook on future work in chapter 6.

Chapter 2

Background

The work presented in this document builds on previous work. The following sections present notable related work in dialogue summarization and document visualization, together with background knowledge, which aims at helping the reader to understand the rest of this document.

2.1 Related Work

Since previous related work contains a variety of domain- and application-specific research, the overview presented is grouped into specific sections:

- Text summarization, which covers general research in the field of single document summarization.
- Dialogue summarization, focusing on multi-document summarization, specifically within the domain of meetings.
- Document visualization, showing currently applied techniques and possibilities.

2.1.1 Text summarization

With the breakthrough of "Attention Is All You Need" [Vaswani et al., 2017b], many subsequent researchers based their models on the introduced transformer architecture.

In the domain of **extractive summarization**, [Wang et al., 2019] shows a solution to dealing with model generalization on data that belongs to unseen fields. They introduce document categories such as sports or business, by which the different data distributions can be classified and provide the Multi-SUM dataset to provide a good multi-domain testbed.

At the same time, within a large-scale evaluation of published models on newspaper summarization, a key factor for successful extractive summarization

is found by [Zhong et al., 2019]. They compared two different usages of the BERT [Devlin et al., 2018] architecture on the CNN/ DailyMail dataset [1]:

1. Feeding each sentence to obtain sentence encoding
2. Feeding the entire article to BERT and obtaining sentence representation through mean pooling

With the latter performing significantly better, they showed that the positional relationship between the sentences has to be present in the encodings to leverage the full potential of the transformer architecture.

Another way of dissembling text into summaries is by using syntactic compression. The idea is to have rules to remove non-key information within a sentence. Examples of such rules used in [Xu and Durrett, 2019] are appositive noun phrases, relative/adverbial clauses, and content within parentheticals. The authors combine a neural extraction model to score sentences from the document with a syntactic compression module to achieve robust performance compared with other state-of-the-art CNN/ DailyMail dataset models.

A different framework utilized in [Wang et al., 2020a] is the Graph Neural Network. In their approach, the graph consists of two nodes: basic semantic nodes (words, concepts) and supernodes (phrases, sentences, and documents), which can establish relationships between each other via basic nodes. The graph updates the nodes during training via Graph Attention Network [Veličković et al., 2017], and the sentence node representations are extracted to produce summaries. Shortly after this study and built on the same Graph Framework is [Jia et al., 2020], which proposes a Hierarchical Attentive Heterogeneous Graph for Text Summarization (HAHSum). They outperform previous extractive summarizers by introducing a redundancy layer responsible for spotting redundancy dependencies between sentences and modeling different levels of information, including words and sentences.

In the domain of **abstractive summarization**, promising research has been done in the last couple of years. For example, with a hierarchical RNN encoder/decoder structure [Cohan et al., 2018] publish the first model for single, long-form documents (specifically research papers). However, in a later study [An et al., 2021], the authors noted a critical point, namely that most scientific papers are full of uncommon domain-specific terms, which makes it difficult for the model to understand its true meaning. They propose a citation graph-based summarization model, which enriches the information of the source paper with references.

Another relevant subfield of abstractive summarization is query-based summarization. It aims to create a brief, organized and informative summary for

[1] https://github.com/abisee/cnn-dailymail

a document with the specifics described in the query. In addition to creating a large-scale query-focused summarization dataset (WikiRef), [Zhu et al., 2019] implements a BERT-based model where the query and document are flattened and concatenated together as a sequence input. As vector representations, they are fed into the scoring and selection layer to rank the sentence by its relevance to the query and salience to the document.

One problem with this approach is that the query is a static representation. As an improvement, [Nema et al., 2017] introduced a query attention model, which learns to focus on different portions of the query at further time steps. In addition, they also tackled the challenge of the issue of repeating phrases in summary with a "diversity-based attention model." As the name reveals, the diversity model ensures that the current context vector is diverse w.r.t to the previous context vector; this discourages repetition in the generated summary.

While single-document summarization has some applications, the need for a solution to a multi-document setting has become ever more prevalent in recent years. Introducing query-based summarization in the multi-document setting [Baumel et al., 2018] shows the first solution as an iterative method to embed abstractive models within a multi-document query-focused summarization. Furthermore, by first sorting the input documents by their overall TF-IDF cosine similarity to the query and then iteratively summarizing them, they also show how the summary length can be explicitly controlled.

With introducing the topic of multi-document settings, it is crucial to look at this specific field of summarization since it has gained some traction in the last few years.

2.1.2 Dialogue summarization

Dialogues in multi-party meetings differ widely from traditional single documents. For example, the input often consists of ill-formed text fragments (utterances) instead of grammatical, well-segmented sentences. On top of that also, additional noise can be introduced through ASR transcription and segmentation.

One of the first approaches of combining multiple previous published models into a fully unsupervised end-to-end meeting summarization framework is shown in [Shang et al., 2018]. Their pipeline consists of the following:

- Text preprocessing: Reducing unigrams and bigrams to single terms.
- Utterance community detection: TF-IDF weighting of utterances, reduced with LSA and clustered with k-means. The goal is to group utterances that a typical abstractive sentence should summarize.
- Multi-sentence compression: Word importance scoring and graph building via word co-occurrence network [Tixier et al., 2016].

- Budgeted submodular maximization: To select a subset of abstractive sentences that are within the maximum size allowed.

They show that their approach outperforms all of the existing baselines on the AMI [2] and ICSI [3] corpus.

The researchers in [Zhu et al., 2020] take it further by leveraging the encoder-decoder transformer architecture. As a novel contribution, they introduce cross-domain pretraining by collecting summarization data from the news domain and converting them into the meeting format: Groups of several news articles from a multi-person meeting, and each sentence becomes a speaker's turn. For simulating a mixed order of speakers, the turns are shuffled. To incorporate the role of each speaker, they train a role vector for each meeting participant to represent the speaker's information during encoding. In addition, a hierarchical structure is introduced with a word-level and a turn-level transformer. The idea here is that the computational complexity is very high during long transcripts, and splitting the transformer into a two-level transformer can incorporate the natural multi-turn structure of a meeting.

What is often overlooked and investigated in [Koay et al., 2020] is the impact of *jargon* in meeting summarization. Jargon is the specialized terminology associated with a particular domain, which might not be understood outside that context. Their study compares models trained with and without jargon, extending the ICSI meeting corpus with human annotations of expressed jargon terms. Their finding is that summarizing with jargon can substantially boost meeting summarization performance (absolute gain of +4.3 % in R-2 F-score). However, it is difficult to obtain and inject this domain terminology in a semi-automatic way.

With the success of BART [Lewis et al., 2020] in neural abstractive summarization, [Koay et al., 2021] extends its framework for meeting and producing meeting minutes. Using a sliding-window approach to break down lengthy transcripts into small local windows lets them find salient content while reducing the complexity of processing long documents. On the ICSI dataset, they achieve the best result with a window size of 1024 (large context window) and a stride of 128.

Especially in meetings, users might be interested in different facets of the meeting. With QMSum [4], a new query-based, multi-domain meeting summarization dataset was developed, consisting of 1800 query pars with over 232 meetings. In their paper, [Zhong et al., 2021] also provides a baseline for future work, comparing previous models such as BART [Lewis et al., 2020] and HMNet [Zhu et al., 2020].

[2] https://groups.inf.ed.ac.uk/ami/corpus/
[3] https://groups.inf.ed.ac.uk/ami/icsi/
[4] https://github.com/Yale-LILY/QMSum

The possibility of supervising the granularity is researched by [Wu et al., 2021]. The authors propose a method for abstractive dialogue summarization and simultaneously enable granularity control. The two stages of their approach are as follows: First, summary sketches containing information about user intent and essential key phrases are created. Then, summarization fragments based on the summary sketches are generated to represent the dialogue's summary. With this, they achieve state-of-the-art performance on the SamSUM [Gliwa et al., 2019], messenger, dataset.

In the recently published paper Tweet Stream Summarization Using BERT [Dusart et al., 2021], the authors propose an approach to "...automatically estimate the appropriate size of the summary to propose at a given time..." in the context of tweet summarization. They have a two-folded pipeline with a flexible summary size to achieve this.

1. Salience prediction - deciding if a tweet should be kept in the event summary (Utilizing the insights from a preceding paper [Li and Zhang, 2021]) having its focus on generating salience predictions on events.

2. Tweet selection - For each tweet in the existing summary, its similarity with the candidate tweet is computed. If the similarity score is lower than a similarity threshold for all the tweets in the existing summary, the candidate tweet is kept for the summary.

For the task of Email summarization, [Zhang et al., 2021] propose an abstractive email thread summarization dataset, EMAILSUM, that contains 2,549 email threads with human-written short and long summaries. Furthermore, they evaluate models such as T5, Oracle, and TextRank and achieve the best results with semi-supervised T5 training. The key finding is that human evaluation reveals that the model fails to understand the sender's primary intention. The roles of different speakers and automatic metrics could be better correlated with human judgment.

2.1.3 Document visualization

An enormous amount of different approaches can be found when researching visualizations of documents. The following selection focuses mainly on visualizations that involve evolving documents over time and having heterogeneous documents, such as conversations between people.

In Visual Document Analysis: Towards a Semantic Analysis of Large Document Collections, [Oelke, 2010] provides a holistic overview of text's characteristics and semantic properties and shows how different visualization methods can support the process of document analysis. In addition, with the help of vocabulary measures (such as specific word frequencies, vocabulary richness, and sentence lengths), the author shows how visual analysis can aid authorship attribution and help spot different topical segments of longer texts. In the visualization example in figure 2.1, the author shows how longer sentences can point to a more formal report segment within a text. In contrast, shorter segments often tend to point to a dialogue.

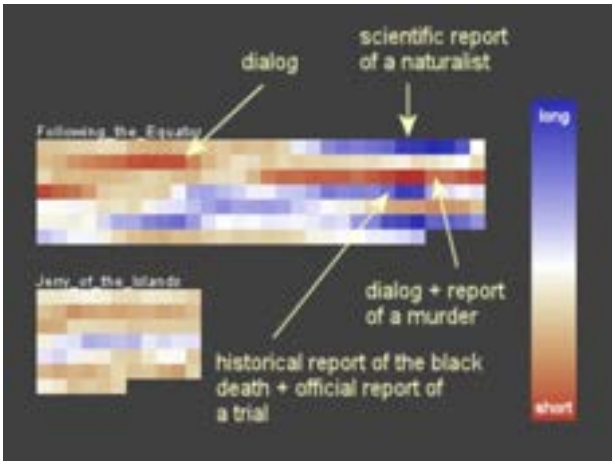

Figure 2.1: "Fingerprints" of two novels that show the different structures of the two novels. The inhomogeneity of the first novel can be explained with different text forms: dialogues, narrative parts, and quoted documents [Oelke, 2010].

Next to the vocabulary measures, the author also proposes a readability measure, defining how difficult it is to read a specific sentence. Additionally, a discrimination measure of overlap terms and methodologies for visually analyzing sentiment and opinion.

With a particular focus on the time dimension, [Liu et al., 2009a] published TIARA around the same time, a tool to interpret and examine the summarized text from multiple perspectives. First, they use a Latent Dirichlet Allocation to extract a set of topics, shown as a layer within a time-oriented visual text summary.

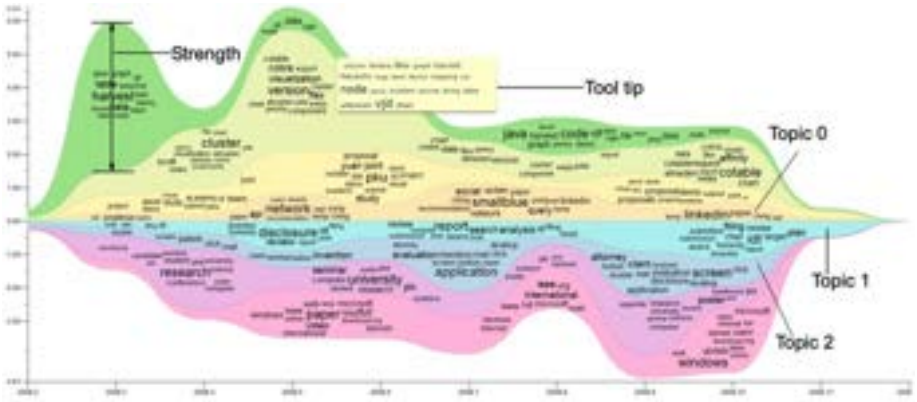

Figure 2.2: TIARA-created visual summary of 10'000 emails. Each layer represents a topic generated by LDA [Liu et al., 2009a].

Then, as a validation scenario, they design a set of email analysis tasks requiring users to answer specific questions using email correspondence between two people. They report that according to their evaluations, the visualization tool was favored by users, especially for more complex tasks.

Building upon the previous "topic stream" visualization, [Dörk et al., 2010] extends the application for a visual backchannel of large-scale ongoing conversations on Twitter. They integrate the topic stream with a people spiral, representing participants and their activity, and an image cloud, encoding the popularity of event photos by size.

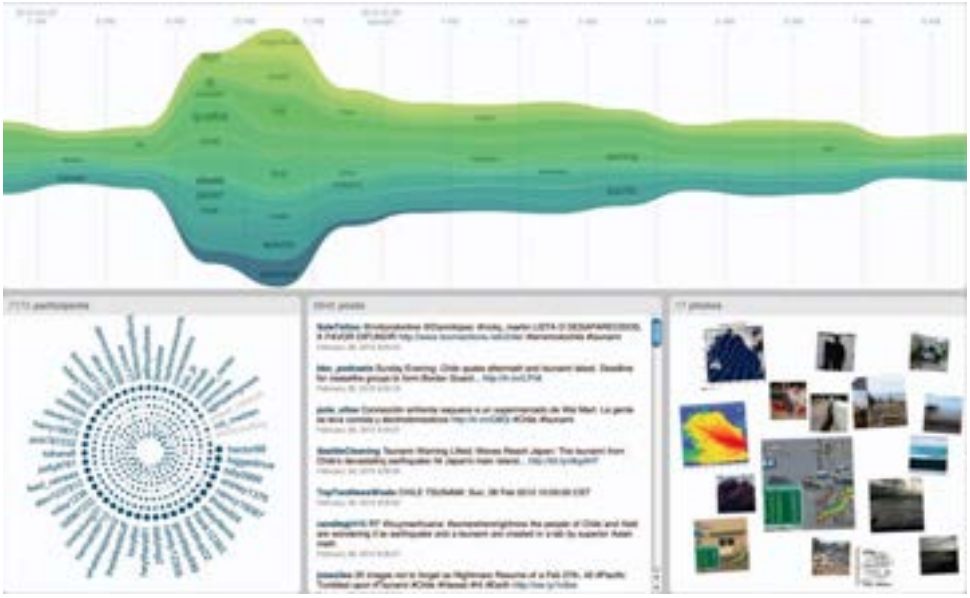

Figure 2.3: Visual backchannel interface representing Twitter posts. Created by [Dörk et al., 2010].

Twitter has been increasingly used for exchanging knowledge and thoughts about events across different parts of the world. Identifying when and where an idea is dispersed is the topic of [Cao et al., 2012]. They propose a novel visualization design, Whisper, based on a sunflower metaphor, whose seeds are often dispersed far away. Social media responses are summarized based on how tweets were retweeted by a group of users, tracing sentiments and retweets on a hierarchical layout.

The design is as follows: The dots in the sunflower's center represent tweets about topics of interest (*topic disc*). The sunflower florets' lines represent the *diffusion pathways*, tracing the path from the information source tweet to different groups of users who retweet the information. *User groups* are then represented by cluster icons at the end of the florets.

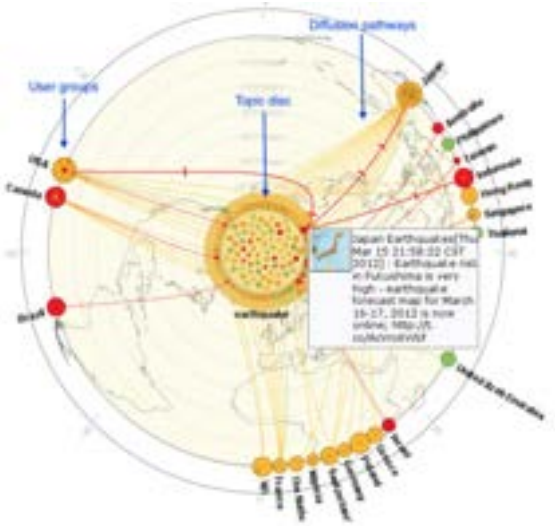

Figure 2.4: Visualization of Whisper, showing diffusion of information on Twitter regarding an earthquake and a series of aftershocks and tsunamis [Cao et al., 2012].

As another tool for visualizing Twitter data, [Humayoun et al., 2017] focuses on a new aspect of analyzing people's reactions to a particular event or product. With TextVis, they provide a method to not only analyze keywords on their frequency but also the relations between them based on their co-occurrence. Furthermore, the visualization uses a chord diagram to deal with cluttering that might occur when multiple relations are associated with the underlying keywords.

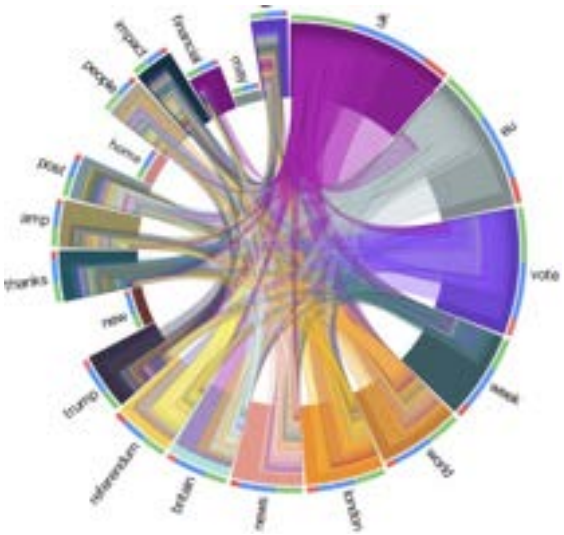

Figure 2.5: TextVis visualization with a chord diagram based on the word similarity relations [Humayoun et al., 2017].

Still, in multiple document collections but with another specified task, [Lee et al., 2009] investigates the problem of reading and exploring response messages to blogs. As more than 10'000 responding messages are registered on a well-known blog daily, it is hard for a user to locate helpful comments among unrelated comments and ad/spam messages. To tackle this challenge, they design a tool (TRIB) for visualizing bloc articles, considering the semantic weight between the subject article and corresponding comments.

Identifying relevant and essential papers can sometimes be difficult when researching an academic topic. However, when following references and citations, there is a great chance to end up with a vast collection of papers. With PaperVis, [Chou and Yang, 2011] set the goal of making literature reviews easy. They arrange papers as a node-link graph to visualize their complex citation-reference structures. To arrange the papers within the network, they define the terms *relevance*, amount of papers they have co-referenced, *level*, a measure of the occurrence of citations for a selected paper, and *importance*, being the percentage of citations within other papers.

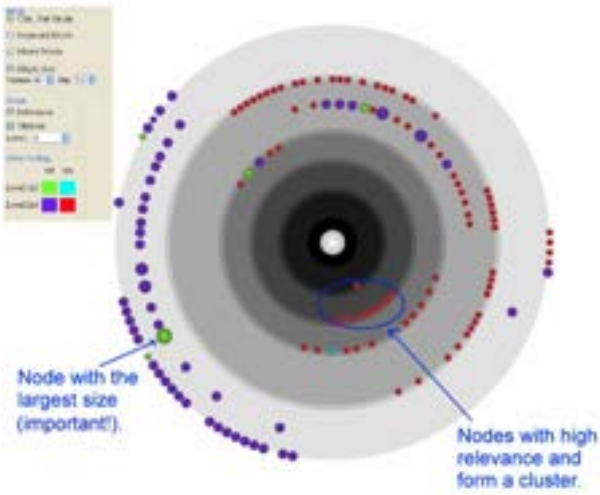

Figure 2.6: Primary visualization of the PaperVis tool [Chou and Yang, 2011].

[Liu et al., 2015] has the focus of their application in the same domain of academic papers and publications. They try to identify lead-lag relationships (defining the order in which topics arise among different corpora) in the context of a specific topic. In the presented tool TextPioneer, they first extract topics from multiple corpora with a Dirichlet process model and derive a hierarchy from organizing them. Then every document gets the topic assigned with the highest probability value. Multiple perspectives of the results are then visualized in a hybrid tree visualization and a ladder-like visualization.

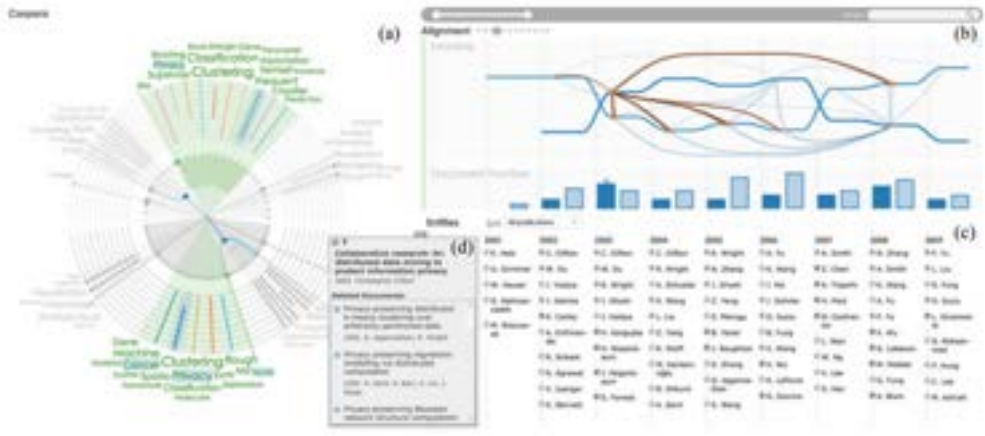

Figure 2.7: The lead-lag relationships for academic publications in data mining: Corpora (a) shows publications lead proposals in the selected area. Alignment (b) shows the lead-lag evolution of the highlighted topic, "Privacy." Entities (c) show the author's information, and document snippet (d) enables the detailed examination of the document [Liu et al., 2015].

With TopicPanorama, [Liu et al., 2014] combine many of the previously mentioned methodologies for analyzing relevant topics. They aim to solve the following tasks with their tool:

1. Obtaining an overview of relevant topics. They achieve this by integrating multiple topic graphs to form a complete visualization of relevant topics based on their content and relationships with each other.

2. Examining each source's common topics and specific topics with a level-of-detail visualization that places common parts near the area of each corresponding source.

3. Examining correlations between topics and exploring the entire picture at different levels of granularity. By leveraging a topic graph, the user can quickly get an overview of the topic while gradually zooming into the detailed context.

4. Analyzing temporal patterns of the matched topics by incorporating lead-lag analysis into the visualization.

In different case studies, they describe the potential application scenarios of the tool by analyzing news media impacts in the public health sector and can support analytical needs in the public relation sector.

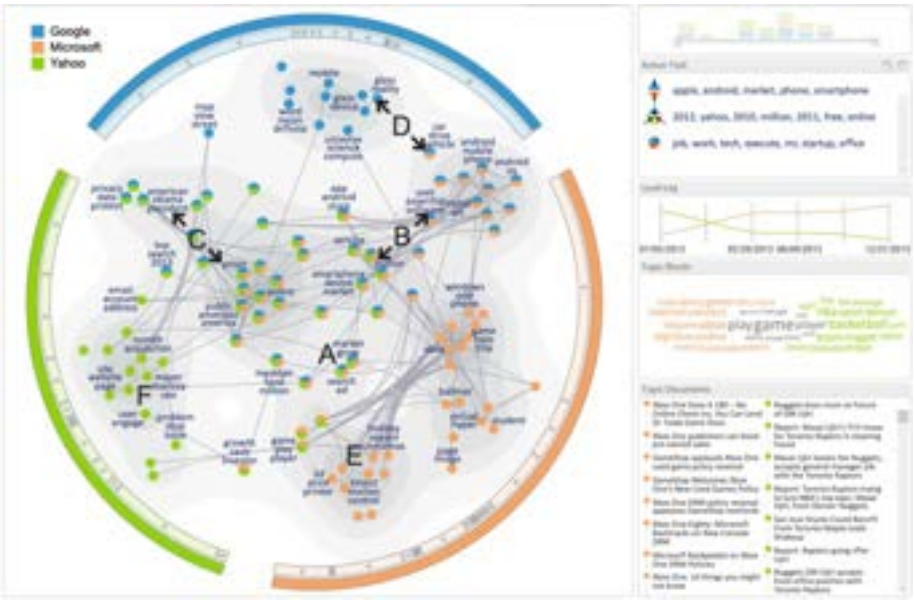

Figure 2.8: TopicPanorama visualization of topics related to Google, Microsoft, and Yahoo: Graph visualization, uncertainty filter, lead-lag analysis, topic cloud, and included documents explorer [Liu et al., 2014].

The academic research could be more extensive when searching for visualization examples focusing on discussions and meetings with additional summarization capabilities. Also, with commercial tools such as [Talkwalker, 2022], [Repustate, 2022], and [Vontage, 2022] that can use ASR input for their analysis, they cannot fully summarize the conversation apart from displaying keyword frequencies. In addition, the explorative aspects that some previously mentioned visualizations offer are very limited when the data is more unstructured, such as from meetings and discussions.

Within the goal to explore and discover new methods to visually analyze this challenging dialogue language, in the next section, the technical terminology is introduced that covers the existing models and newly developed methods applied within the experiments of this thesis.

2.2 Preliminaries

This section explains the Natural Language Processing (NLP) techniques that drive the design and experiments of this thesis. The commonality among most is to recognize the main patterns of information and features from a text corpus and help the user to form decisions from them.

2.2.1 Semantic analysis

Semantic analysis is an umbrella term covering a branch of linguistics within the framework of NLP, interpreting language structures through text analysis. It aims to automatically interpret sentences, paragraphs, or even documents by identifying their grammatical structure and relationships between individual words in a given context. The extracted meanings can be utilized for various tasks, crucial in many ML tasks such as translation or summarization, where the accuracy of the summary highly depends on the machine's ability to understand the dependencies of the language data. The three main parts of semantic analysis that modern automated systems incorporate into their method are:

1. *Lexical semantics*, for understanding relationships between textual entities (such as Synonyms, Homonyms, and Hyponyms).

2. *Word sense disambiguation*, dealing with the definition of which sense a word is used in, based on its context ("Is orange referring to a fruit or a color?").

3. *Relationship extraction*, detecting relationships between two or more entities within a text ("Shakespeare is the *author* of Hamlet").

Different semantic extraction or classification techniques can be applied depending on the information that needs to be extracted. The following parts will cover some of the main methods used in combination with a visual representation layer for the context of this thesis.

Sentiment analysis

The idea of sentiment analysis (also called opinion mining) is to understand the author's emotional state of a text, depending on the context, it is commonly framed by five elements, defined in [Liu, 2010] as *opinionated text model*:

- Opinion holder: Entity that expresses the opinion.
- Object: The target entity of the opinion.
- Aspect: Specific target feature about which the opinion is stated.
- Type / Polarity: Expressing the orientation of the opinion. Most commonly positive, negative, and neutral, along with an indication of strength.
- Time: Time when the view was expressed by the opinion holder

This prerequisite shows that sentiment analysis is a challenging problem because even identifying each piece of information is already very difficult, let alone finding all five and matching them. As an additional challenge, a sentence may not explicitly mention some pieces of information, but they are implied due to pronouns, language conventions, and context. Depending on the task, however, only some of the full quintuple of information needs to be discovered. Some may be known or even not needed at all. For example, in movie reviews, the object (movie) evaluated in each review, the time of submission, and the opinion holder are all known as review sites usually store such information.

Sentiment analysis can be performed on different extraction levels: The aspect or feature level (what aspect of the object the particular user likes or dislikes), the sentence level, and the document level. There are three approaches to performing this analysis: Lexicon-based, Machine-learning-based, and hybrid approaches.

Word	Score	Description
care	+2	verb
good	+2	adjective, verb
ordinary	0	adjective
complicated	-3	adjective
most	+100%	intensifier
little	-50%	intensifier

Table 2.1: Dictionary-based resource example.

Historically, the primary go-to techniques for sentiment analysis were **Lexicon-based**, divided into

two approaches: dictionary-based and corpus-based. In the dictionary-based method, the classification uses a lexical resource of terms. An example is SentiWordNet [Baccianella et al., 2010], a predefined dictionary with annotated synsets (set of word synonyms) according to notions of 'positivity,' 'negativity,' and 'neutrality.' An example displaying the scoring mechanism of this approach can be seen in 2.1.

On the other hand, in the corpus-based approach, the analysis does not rely on a predefined rule set. Instead, it is conducted by statistical techniques such as hidden Markov models (HMM) [Odumuyiwa and Osisiogu, 2019] or k-nearest neighbors (k-NN) [Kaur et al., 2018].

In the Machine-learning-based category, the techniques can be distinguished into "Traditional" and Deep Learning methods. The traditional methods cover models such as Support Vector Machines (SVM) [Ahmad et al., 2018] and the Naive Bayes Classifier [Dey et al., 2016]. The input to those models can vary between lexical features, adjectives, adverbs, or parts of speech. In most cases, Deep Learning models provide better results since they can consider more context, from the sentence to the document level, when performing the classification task. A recent comparative study from [Nandwani and Verma, 2021] shows this, stating that in some cases, "...traditional Machine Learning models fail to extract some implicit features or aspects of the text and that in situations where the dataset is vast, the Deep Learning approach performs better than Machine Learning."

For the experiments of this thesis, the sentiment analysis model is DistilBERT [Sanh et al., 2019], which is based on a lightweight Neutral Network / Transformer architecture, from which its general principles will be explained in more detail in section 2.2.2. Since sentiment classification is just one small part of the overall solution, the benefit of using DistilBERT is that it is swift in computation, compared to other Transformer architectures, while retaining most of the language understanding capabilities of larger models. The current model used is fine-tuned on Glue [Wang et al., 2018] and the SST2 [Socher et al., 2013] datasets. At inference, the text is fed at document level into the model, returned are the labels $Negative, Positive$, together with their probabilities $p \in [0, 1]$.

Named Entity Recognition

Named Entity Recognition (NER) identifies essential entities from a given text. Some of the most common categories are:

- Person
- Organization
- Location

NER can be described as a sequence tagging task, where the model receives a set of sentences and returns a list of predicted tags:

> Given a set of set of documents $D = \{d_1, d_2, ..., d_n\}$ where each document d_i consists of a word sequence $[w_1, w_2, ...w_n]$, find the entity label $y_i \in \{\text{PER, ORG, LOC, MISC, 0}\}$ for each word w_i, with a scoring function $f(y_1, ...y_n, w_1, ...w_n)$, defining how fit a labelling sequence $[y_1, ...y_n]$ is to a given word sequence.

Over the past decades, various techniques have been proposed to solve the NER task. The initially proposed methods started with hand-crafted rule-based linear models, intended to fit a specifically structured text corpus [Jacobs and Rau, 1993] and evolved with technology to more generalizable supervised learning methods such as Decision Trees and Support Vector Machines [Asahara and Matsumoto, 2003]. More recently, Neural Network based approaches have been significantly successful when leveraging pre-trained word embeddings. Introduced by [Mikolov et al., 2013] as a highly granular vector representation of words, the bag-of-words, and skip-gram models, are still highly utilized today, with the great benefit of this method being the possibility of linear semantic inference ("Paris" - "France" + "Italy" = "Rome").

Today, the typical NER models are built on this idea of pre-trained embeddings within the encoding-decoding framework, in which the semantics of words are embedded into the encoder, and the decoder adopts the word representations to predict their tags. Since word context within a sequence contains important semantics, mostly deep Neural Network architectures like BILSTM and Transformer [Yan et al., 2019] are used to capture the temporal information of the text.

Keyword extraction

Keyword extraction is an NLP technique of automatically extracting essential terms, phrases, or words from a text to represent the document concisely. It assists in the identification of crucial issues and statements. It can be applied in several applications, such as the summarization of documents, sentiment analysis, and automatic text clustering/indexing. Typically, methods for keyword extraction can be grouped into two categories: Supervised and unsupervised. The unsupervised methods use statistical features of words to extract keywords utilizing a scoring or weighting metric. Techniques include:

- Use of characteristics of word frequency, position, and occurrence, as well as capitalization such as the YAKE algorithm [Campos et al., 2018].
- Language models for scoring phrases and informativeness within a single score [Tomokiyo and Hurst, 2003].

- Clustering approaches such as hierarchical or spectral clustering to ensure the semantical coverage of the whole document [Liu et al., 2009b].
- Graph-based ranking algorithms such as TextRank [Mihalcea and Tarau, 2004].

Among the unsupervised methods, graph-based approaches are the most commonly used. They transform the words in a document into a graph, where each node represents a term or feature extracted from the document, and the edges represent the relationship between them. The ranking is then done for each word recursively based on global information drawn from the entire graph.

For supervised methods, keyword extraction is often formulated as a binary classification problem [Hulth, 2003], where words in a document are classified as a keyword (1) or no keyword (0). Formally, the supervised keyword extraction task can be defined as:

> Given a set of set of documents $D = \{d_1, d_2, ..., d_n\}$ where each document d_i consists of a word sequence $[w_1, w_2, ...w_n]$, each word has an assigned label such that
>
> $$f(g(d_i, w)) = \begin{cases} 1 & w \text{ is a keyword} \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$
>
> and $g(x)$ denotes the function for generating features for a given d_i and w.

It requires a considerable amount of training data but usually outperforms unsupervised methods. In traditional supervised learning methods such as Decision Trees and Support Vector Machines, handcrafted feature vectors with lexical features and statistical information are used to train the model. This method's drawback is handling out-of-vocabulary words, which can usually only be handled at inference if a word appears in the training corpus.

As with the previously mentioned methods for semantic analysis, advances in Deep Learning have made Neural Network based approaches more common for keyword extraction. The main benefit of those approaches is that they remove the process of manual feature engineering by automatically discovering the relationship between the text input and the keyword output. Furthermore, with recent studies such as [Zhang et al., 2020], the authors show the potential of this method to outperform other methods significantly.

Text summarization

Text summarization is a method of automatically decreasing the size of a document (or sets of documents), condensing the source text into a smaller and

compact version while preserving its overall meaning and having minimum loss in the overall information content. The general problem can be stated as follows:

> Given a set of documents $D = \{d_1, d_2, ..., d_n\}$ where n is the number of documents and every document d_i consists of a set of sentences $S_{d_i} = \{s_{1d_i}, s_{2d_i}, ..., s_{nd_i}\}$, find the subset of sentences $s \subset S$ $S = \{S_{d_1}, S_{d_2}, ..., S_{d_n}\}$, that covers the different topics in the document collection while reducing the redundancy within the summary.

Generally, a document contains information centered around a central theme, covering different aspects. As such, a generated summary should cover those different aspects or subtopics as much as possible. The summarization methods can be split into two categories: extractive and abstractive. In addition, each has sub-categories such as single- or multi-document, query-focused, or supervised and unsupervised methods.

Extractive methods work by picking a subset of existing words, phrases, or sentences in the reference text to form the summary using statistical features. Within the generated summary, no additional generated words are introduced. Usually, the extraction is performed in three steps [Sonawane et al., 2019]:

1. Transformation of the original text document, representing the document in the form of its elements like paragraphs, sentences, and tokens. Additionally, pre-processing methods like stop word removal and stemming are performed in this step.
2. Sentence scoring, using a ranking algorithm to score relevant sentences
3. Sentence selection, generating a representative summary using the previous steps.

Abstractive summaries can be described as a compressed version of the text, where source concepts and ideas are reinterpreted and presented in a different form. It requires language generation capabilities to generate novel words and phrases not found in the source document. Abstractive summarization is more flexible than extractive methods, making it more likely to produce fluent and coherent summaries. However, it can also be more challenging due to its unconstrained nature. Some challenges can include generating hallucinated content [Kryscinski et al., 2019] containing factual errors and controlling the produced summaries. Some proposed methods to reduce those issues, such as guidance signals, constrain the summary to deviate less from the source document and allow for controllability through user-specified inputs.

In the early days, extractive summarization was the more prevalent, utilizing rule-based approaches or statistical methods like bag-of-words or TF-IDF

extracting sentences containing high-frequency words. However, in recent years, mainly abstractive methods have been preferred due to their ability to generate new sentences. With Neural Networks allowing an end-to-end framework for natural language generation, success has been witnessed on tasks like machine translation, image captioning, and abstractive sentence summarization [Chopra et al., 2016]. Lately, sequence-to-sequence models with attention mechanisms such as the Text-To-Text Transfer Transformer (T5) [Ramesh et al., 2022] proved to produce state-of-the-art performance when being finetuned with task-specific datasets.

2.2.2 Architectures

The following section covers the main model architectures used within the different semantic analysis tasks to extract the different information components from the text corpus.

Recurrent Neural Networks

Most initial proposed methods used the Recurrent Neural Network (RNN) encoder-decoder architecture in sequence-to-sequence tasks such as machine translation or summarization. The encoder reads and encodes a source sentence into a fixed-length vector while the decoder outputs the target language or summary from the encoded vector. The whole encoder-decoder system is jointly trained to maximize the probability of a correct translation given a source sentence.

Generally speaking, RNNs are Neural Networks specialized in processing a sequence of values. This processing is mainly made possible due to parameter sharing, which is particularly important when the same information occurs at multiple timesteps within the sequence. Within RNNs, each output is a function of the previous outputs, produced using the same update rule.

During training time, the network uses the internal state $h^{(t)}$ to map the task-relevant aspects of the input sequence $\{x^{(t)}, x^{(t-1)}, ..., x^{(1)}\}$ to a fixed length vector. One example of this could be in the task of predicting the next word in language modeling, where it is unnecessary to know the whole sequence's information. Instead, a subset is enough to predict the rest of the sentence.

As described in [Goodfellow et al., 2015], the forward propagation of the network begins with the initialization of the hidden state $h^{(0)}$. Then, for each time step, the hidden states are calculated using an activation function σ such as $relu$ or $tanh$, the weight matrices U and H, and the bias vector b:

$$h^{(t)} = \sigma(Ux^{(t)} + Wh^{(t-1)} + b) \tag{2.2}$$

Next, the output values at time t are calculated with the hidden-to-output weights V and the bias vector c:

$$o^{(t)} = Vh^{(t)} + c \tag{2.3}$$

Lastly, the target predictions $\hat{y}$ are computed through a scoring function such as $softmax$ and the outputs $o^{(t)}$:

$$\hat{y}^{(t)} = softmax(o^{(t)}) \tag{2.4}$$

In the case of an RNN that maps an input sequence to an output sequence of the same length, the total loss for a given sequence of x values would then be the sum of losses over all time steps.

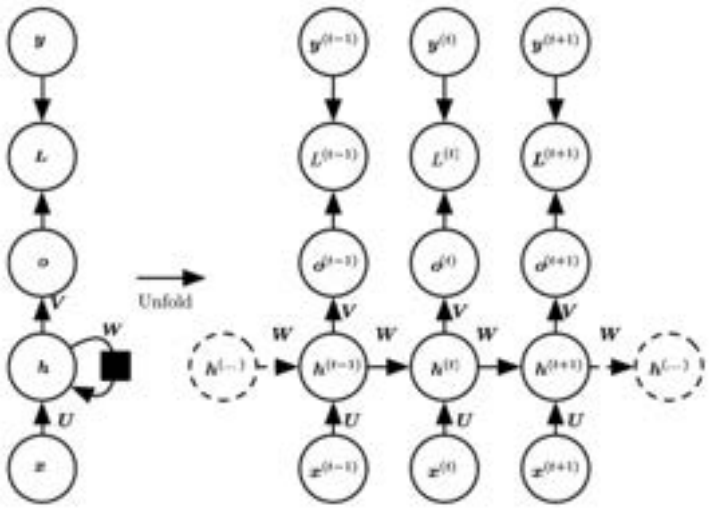

Figure 2.9: The computational graph of an RNN, mapping a sequence of input values x to a sequence of output values o. The loss function L computes $\hat{y}$ and compares this to the target y. The hidden states are parametrized by matrices U, W, and V, defining the weights between the input, hidden states, and output [Goodfellow et al., 2015]

.

Encoder-Decoder

However, the input and output sequence is not the same length for many NLP tasks. The encoder-decoder architecture solves this issue. In this type of RNN, the aim is to find a representation, also called context C that summarizes the input sequence $X = \{x^{(1)}, ..., x^{(n_x)}\}$.

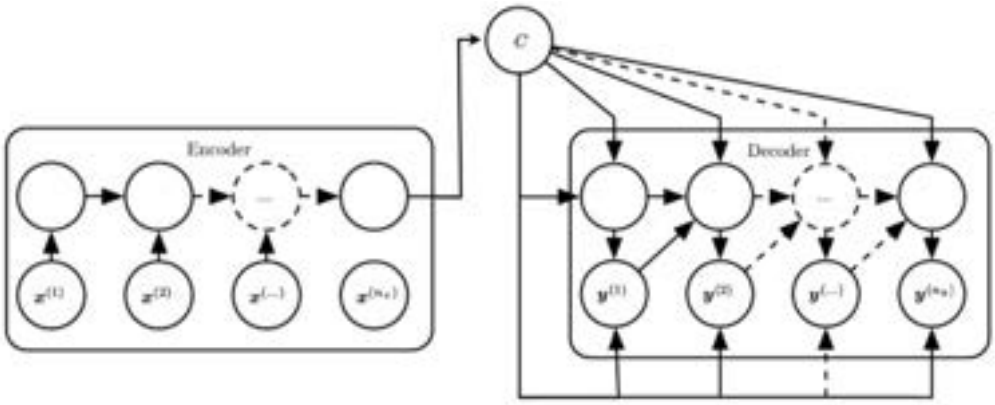

Figure 2.10: The encoder-decoder RNN architecture for generating an output sequence of y values with an input sequence of x values. The final state of the encoder is used to compute the context C, representing a semantic summary, serving as input to the decoder [Goodfellow et al., 2015]

.

To train this architecture, the encoder and decoder are trained jointly to maximize the average probability $P(y^{(1)}, ..., y^{(n_y)}|x^{(1)}, ..., x^{(n_x)})$, over all sequence pairs of x and y in the training set. The last state of the encoder h_{n_x} is typically used as representation context for the input sequence, which is provided as input to the decoder.

Attention

One limitation of this approach when dealing with long sequences is having too small of a context C that can not correctly represent the whole input sequence. As the predecessor to the modern encoder-decoder architectures we have today, [Bahdanau et al., 2014] proposed a concept for a variable-length context vector C. In addition, they introduced the first attention mechanism responsible for associating parts of the sequence C to the output sequence of the decoder. With this approach, the information spread throughout the input sequence can be selectively retrieved by the decoder, thus relieving the encoder from representing all information of the source sentence into a fixed-length vector.

In the approach of [Bahdanau et al., 2014], a Bidirectional RNN is used to read the forward hidden state sequence $\{h^{(1)}, ..., h^{(t)}\}$, as well as reversing the order to generate the backward hidden states $\{h^{(t)}, ..., h^{(1)}\}$. Then an annotation for each word x_i is generated by concatenating the forward and backward hidden state, which serves within an alignment model, scoring how well the inputs and outputs match. The context vector C is computed as a weighted sum of these annotations.
Nevertheless, there is still a critical limitation to this approach. Each sequence has to be treated sequentially, one element at a time. As a result, both the

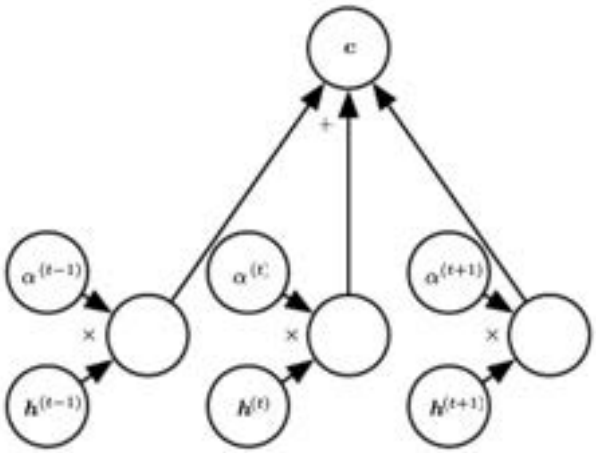

Figure 2.11: A simplified visualization of the attention mechanism. The context vector C is constructed through a weighted average between the hidden units $h^{(t)}$ of the Neural Network and the annotation weights $a^{(t)}$ [Goodfellow et al., 2015]

.

encoder and decoder must wait to complete the t steps to process the $t+1$ step. This prerequisite means that dealing with a large corpus can be time-consuming and computationally intensive. The following architecture addresses this problem by relying entirely on the attention mechanism and proposing a way of parallel processing the input sequences.

Transformer

Introduced with the research of [Vaswani et al., 2017a], transformer-based methods have been the state-of-the-art architecture for many NLP problems. In the previously described attention mechanism, attention is used to form an intermediate state between the encoder and decoder.

The difference in the transformer architecture is that it uses an improved self-attention mechanism, an internal state between layers, deciding which part of the output from the preceding layer to focus on.

Instead of using a fixed embedding for each token, self-attention produces for every input sequence a new sequence of embeddings by computing a weighted average:

> Given a sequence of n input vectors with length k $X = \{\vec{x_1}, ..., \vec{x_n}\}$ and their corresponding output attention vectors $A = \{\vec{a_1}, ..., \vec{a_n}\}$ with the same dimension, the self-attention operation is a weighted average between the input vectors X and the attention weights W

$$A = W \cdot X \tag{2.5}$$

And the attention weights W are calculated as a row-wise softmax.

$$W = softmax(X \cdot X^T), \qquad w_{ij} = \frac{\exp(\vec{x_i} \cdot \vec{x_j}^T)}{\sum_{j=1}^{n} \exp(\vec{x_i} \cdot \vec{x_j}^T)} \tag{2.6}$$

As the dot product between x and its transpose produces values between negative and positive infinity, the softmax function maps the values between 0 and 1 and ensures that $\sum_j w_{ji} = 1$.

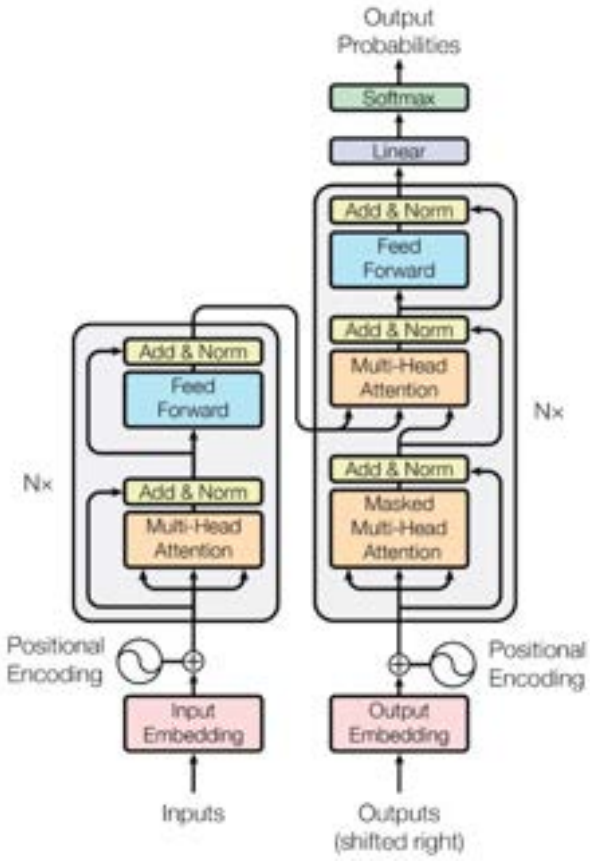

Figure 2.12: The transformer architecture, with the encoder and decoder composed of N stacks of identical layers [Vaswani et al., 2017a]

In the actual transformer self-attention, the input representation of each word vector is more granular. It consists of the *query*, *key*, and *value* vector, derived by applying a linear transformation to the original input vector, illustrated in 2.13.

In matrix notation, the attention calculation can be written as:

$$Attention(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \tag{2.7}$$

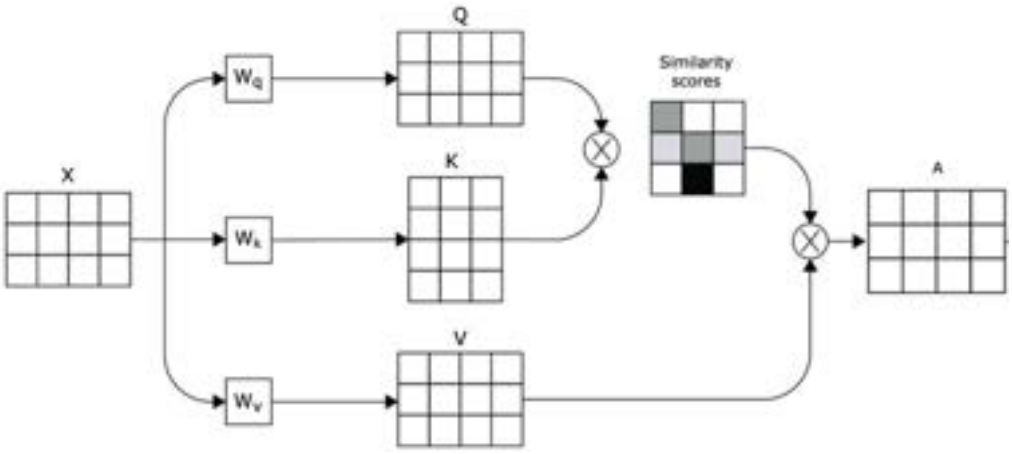

Figure 2.13: The Attention mechanism with query, key, and value transformations

.

The transformation is multiplying x with the weight matrices (W_q, W_k, W_v), the values of which are learned during the training process:

$$\vec{q_i} = W_q \cdot x_i \qquad \vec{k_i} = W_k \cdot x_i \qquad \vec{v_i} = W_v \cdot x_i \tag{2.8}$$

Then similarity scores s are calculated between the query and key vectors, representing commonalities (similar vectors will have a larger dot-product than vectors that share no overlap) and scaled by the input dimension k, which stops to large input values. The attention weights are then computed by normalizing the similarity scores with a softmax function:

$$s_{ij} = \frac{\vec{q_i}^T \cdot \vec{k_j}}{\sqrt{\mathrm{k}}} \qquad w_{ij} = softmax(s_{ij}) \tag{2.9}$$

Once the output weights are computed, they are multiplied by the value vector $\vec{v}$ to obtain the updated output attention representation:

$$\vec{a_i} = \sum_j w_{ij} \cdot \vec{v_j} \tag{2.10}$$

Several attention layers are combined and run in parallel to increase the learned projections of the self-attention mechanism, named by the term *Multi-Head attention* [Vaswani et al., 2017a].

In the multi-head attention block consisting of h parallel layers, each attention head has its weight matrices W_q^i, W_k^i, W_v^i. For the input vector $\vec{x_i}$, every head produces a different output vector $\vec{a_i}^i$. To reduce the dimensions back to k, they are concatenated and passed through a linear transformation. By chunking the

input vector into smaller batches, the attention mechanism with multiple heads in parallel is roughly as fast as applying a single self-attention mechanism with the full input vector.

For most transformers, this multi-headed attention is wrapped together with other components to form a repeatable block, as illustrated in 2.12. In this block, the Multi-Head attention is followed by a normalization layer, a simple two-layer fully connected feed-forward Neural Network, which processes each embedding independently, followed by another normalization layer. Both encoder and the decoder have the same building blocks, the main difference being that the decoder has two attention sublayers.

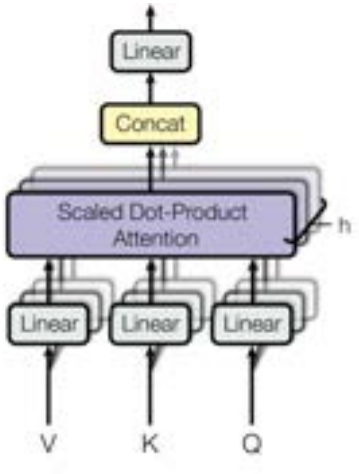

Figure 2.14: The Multi-Head attention block [Vaswani et al., 2017a].

The decoder's first *Masked Multi-Head Attention* layer ensures that the generated tokens at each timestep are only based on the past outputs and the current predicted token. The second *Encoder-Decoder attention* layer performs the attention operation with the intermediate representations of the decoder acting as queries. Using the attention layer on the decoder output serves the purpose of learning how to relate tokens from two different sequences, e.g., the source text and its summary, together [Tunstall et al., 2022]. Once the output is generated from the decoder, a softmax function assigns a probability for each token.

With transformers, modeling relationships between words has become easier than ever. Since the proposal of its initial architecture, there have been many variations for different tasks, such as BERT [Devlin et al., 2018] and T5 [Raffel et al., 2019], making it the go-to architecture for many modern NLP applications.

2.2.3 Performance metrics

Since the development of automatic text processing methods, there has been a need to evaluate the generated text that allows for comparison methods. While human evaluation is invaluable for getting the first glance at a model's performance, analyzing extensive text collections is very expensive and can take weeks or months to complete. Especially with models under active development, with daily changes and improvements, this can be a big problem. That is why the need for an automated system for evaluating the performance of NLP techniques

has evolved. The two most prominent are shown in more detail in the following sections.

BLEU

Since its development, BLEU [Papineni et al., 2002] has been one of the predominant ways to measure the performance of machine-produced text, especially in machine translation. Its main benefit is its independence from a specific language and easy computation.

Generally, BLEU compares n-grams of the predicted text with the n-grams of the reference text, counting the number of matches. The comparison is made using a precision score, while matches are position-independent. The scoring metric proposed is a modified unigram precision that accounts for machine translating systems that tend to overgenerate high-probability words. This modification clips the total count of each generated word by its maximum count in the reference sentence and divides the sum of the clipped counts by the total number of generated words. For an entire corpus, the modified precision score p_n is extended to all sentences:

$$p_n = \frac{\sum_C \sum_{n-gram} Count_{clip}(n-gram)}{\sum_C \sum_{n-gram} Count(n-gram)} \tag{2.11}$$

C is a predicted candidate/sentence in the corpus and $n-gram \in C$. This incorporates all n-gram precision scores into one metric; the geometric mean taken for each score p_n, considering precision decreases exponentially as n increases, requiring logarithmic averaging.

$$\text{BLEU} = BP * \exp(\sum_{n=1}^{N} w_n log(p_n)) \tag{2.12}$$

N is the number of $n-gram$ lengths considered, and the brevity penalty BP is a penalizing factor that compares the candidate translation length with the reference text length:

$$\text{BP} = \begin{cases} 1 & \text{if c} > \text{r} \\ exp(1-\frac{r}{c}) & \text{if c} \leq \text{r} \end{cases} \tag{2.13}$$

Where c is the length of the candidate translation and r is the reference corpus length.

ROUGE

As a closely related measure, ROUGE [Lin, 2004] is another measure for evaluating machine-generated text, focusing on recall, while the previously mentioned

BLEU score is more precision-oriented. At its core, ROUGE is an n-gram recall between a candidate sentence and its reference sentences:

$$\text{ROUGE-N} = \frac{\sum_S \sum_{n-gram_n} Count_{match}(n-gram_n)}{\sum_S \sum_{n-gram_n} Count(n-gram_n)} \tag{2.14}$$

n is the length of the n-gram, and $Count_{match}(n-gram_n)$ represents the maximum number of matches between the candidate text and its reference texts. When multiple references are present for one candidate text, the final score is computed by taking the pairwise maximum of each candidate sentence s and its references r_i.

$$\text{ROUGE-N}_{multi} = argmax_i \text{ROUGE-N}(r_i, s) \tag{2.15}$$

The ROUGE-N metric is only one of multiple ROUGE measurements proposed in [Lin, 2004]. There are other variations, such as:

- ROUGE-L, based on the longest common subsequence
- ROUGE-W, a weighted version of ROUGE-L, accounting for consecutive matches between the candidate and target
- ROUGE-S, which calculates skip-bigram co-occurrence statistics, measuring the n-gram matches while allowing for arbitrary gaps

In most studies that evaluate their performance on the ROUGE metric, multiple variations are reported together to display a full picture since they all capture a different aspect of the evaluation.

Sequence labeling evaluation

The usually reported metrics for the sequence labeling tasks, such as Named Entity Recognition, contain a precision, recall, and f1 score. In the model evaluations, the implementation of Seqeval [Nakayama, 2018] is used for computing the scores, which are defined as:

$$Accuraccy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.16}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.17}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.18}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.19}$$

TP represents the positive cases correctly predicted, and TN represents the negative cases correctly predicted. FP and FN stand for the misclassified positive and negative cases.

Chapter 3

Design

This chapter is inspired by the primary goal of building a visualization that presents a text's essential information in the most time-efficient way. First, it gives an overview of the essential components of the visualization layer with their design choices, then shows the different applied NLP techniques used for data preparation and processing.

3.1 Problem definition

With the various available visualizations presented in 2.1.3, finding and defining the "best" one is challenging. As with all designs, it is important to define a problem statement to be solved clearly. For this research, the general problem that is to be solved can be stated as follows:

> The steady increase in texts from meetings and conversations affects how people process information. For example, recollecting critical information in a reasonable amount of time becomes more complex as text increases.

Having this statement in mind, the next important thing to define is the two questions:

1. Who is experiencing this problem?
2. Where is the problem happening?

The first question is a very crucial one. In a typical business environment, the daily flow of text conversations happens on all different stakeholder levels. The critical information can completely differ from someone else's depending on the individual's position. For example, take a meeting where the conversation was transcribed for securing purposes. There might be internal business representatives, technical consultants, and external stakeholders. Here, the business

representative might be interested in a more general summary. Some technical details are probably more important for a specialist, while the external stakeholder might look for specific summaries such as "strategic decisions."

The answer to the second question is probably a vast one. In today's environment, there are countless places where one encounters conversations in text format. It can be a casual chat on social media, a back-and-forth email conversation, messenger group conversations, and transcribed meetings (online or physical). Even though the general problem exists in all those cases, it would require extensive time and resources to cover them in this research. For this reason, the visualization will mainly cover transcribed meeting data, while exploring its expansion for incorporating text collections from other domains remains an open task.

To summarize the main takeaways from the discussion above, the visualization has to be dynamic and interactive based on what information a given user wants to see, including the ability to control:

- **Granularity** (how much context is presented),
- **Dimension** (is the presented information specific to a point in time) and,
- **Entity** (should the visualized information be about a specific speaker or a specific word)

As a constraint, the boundary for the visualization layer is set on meeting transcripts, focusing on this specific domain.

3.2 Design process

Having defined the main structure and points that should be addressed, the next step for this thesis was the process of iteratively designing parts of the solution, gathering feedback, and going back to improving the design. Below, this process is shown, together with the first drafts, focusing on the three main aspects: Control of Granularity, Dimension, and Entity.

3.2.1 First iteration

The first ideas brought to design focused mainly on the time dimension. The idea here is that a given user can manually scroll through a meeting transcript, or even an automatic animation is played at a given speed to minimize the time needed to see all information. Essential words or phrases in the transcript are highlighted, while the overall topic for a given time section is displayed in a prominent position. Figure 3.1 and 3.2 show two example design snippets of this

first draft. Both designs show the full transcript text with the timeline on the vertical axis, mimicking a scroll or chat experience.

Figure 3.1: Visualization design in the first iteration. The transcript is displayed in its detailed form, along with highlighted timestamp information and important words.

The first figure shows the transcript in a more enlarged and spaced layout. In contrast, for the second figure, speaker information is also included, having an assigned color for each speaker and a chat-bubble layout as in most messenger applications.

Figure 3.2: Visualization design in the first iteration. A more messenger-like visualization, displaying the entire transcript with the inclusion of speaker information.

In the next step, self-evaluation, as well as the gathering of feedback, was done. Then, with the primary goal in mind of maximizing the intake of information in the most time-efficient way, the following points for improvement were noted:

1. The quantity of information presented is too large. It is unnecessary to see the entire context simultaneously to get a primary overview of the discussed topic.

2. There is only a limited benefit of knowing when an utterance was made. In the current design, the time dimension is too predominant.

3. The differentiation of speakers is valuable and should be kept in some form.

While solving the problem of controlling the **dimension** aspect, the points above show that the control of **granularity** and **entity** are not addressed with this first draft. Therefore, in the next iteration, proposals are shown to tackle those limitations.

3.2.2 Second interation

The subsequent designs bring some new components into the picture. For one, additional information about the participants in the meeting is shown, and the ability to control what is displayed is done more interactively based on user actions.

In the design proposal, displayed in 3.3, the speakers are grouped in a bubble chart, the size measured by the number of words uttered by a given speaker. The other component is a clickable word-cloud component with the essential n-grams of a given time segment. This component is interactive in that when an n-gram is clicked. The underlying transcript shows the specific utterance where this n-gram occurs.

Figure 3.3: Visualization design in the second iteration. A bubble chart shows individual speaker contributions and a clickable word cloud.

Another design idea was to show even less information on the first view - to present the main keywords and then, based on user actions, expand the specific information that interests the user, seen in figure 3.4.

Figure 3.4: Visualization design in the second iteration. Three different levels of context can be explored interactively based on user actions.

Here in the first expansion level, summary sentences are presented on click of a specific keyword, showing only information about this keyword. Next, cropped utterances involving this keyword are shown on the second expansion level, giving more context to the summary sentence. Finally, the third expansion shows the complete utterance with the neighboring utterances.

In addition to the keyword expansion navigation, components are tested that present additional information about different aspects of the conversation (figure 3.5). The first component shows the topic distribution along the timeline, indicating at what segment of the conversation a given keyword was discussed. The second component shows the sentiment across time, together with an overall sentiment presenting the average mood of the discussion. Lastly, the third component shows a similar bubble chart as in figure 3.3, representing the individual speaker's contribution but also showing at what point any given speaker was talking.

Figure 3.5: Three additional components present different aspects of the conversation—top: topic distribution, middle: overall sentiment, and bottom: speaker contribution.

Before going to the final implementation, another evaluation and feedback round were made for the current design proposals, which brought up the following points:

1. The different levels of expansion depth are valuable as the user can decide which context is necessary to gain the amount of knowledge needed.

2. There is still too much weight on the time dimension, which brings limited value and should only be for specific components, such as sentiment analysis.

3. An additional component that shows the flow of the discussion would be interesting, depicting a sort of intensity (numerous speaker changes in a short interval representing an intense discussion or monologue, while no speaker changes represent a single monologue)

4. General metadata should be added that shows the duration of the meeting and number of participants

With the intent to include this additional feedback in the final solution, the following section shows how the different aspects are incorporated and implemented into a fully functional prototype.

3.3 Final design and implementation

As with many NLP solutions, unique techniques only bring their total value if utilized in conjunction. The final design uses a more dashboard-like layout to consider this, where multiple components can be brought together to provide as much essential information as possible. Figure 3.6 presents this layout, the components of which will be described in more detail, while the data processing and modeling needed for each component will be described in **??**.

Figure 3.6: The final design of the visualization layer, combining multiple components into a dashboard.

The **Metadata** component contains general information about the meeting or discussion. For example, it shows the number of participants, meeting length, and the speaker turns (how many times the speaker changed during the meeting).

In the **Entities** component, the different extracted entities are, on the one hand, presented in a bar chart together with the occurrence of their categories (person, location, organization, and miscellaneous), and on the other hand as scrollable word chips, sorted according to their appearance in time.

The **Sentiment vs. Speaker changes** graphs show two types of information along a time axis (the time axis in the example figure 3.6 is grouped into one-minute time segments). The line chart on top shows the sentiment per time segment, while the bottom chart depicts the speaker changes.

The aim of the **Speaker Network** is to show interactions between the speakers. The number of utterances defines the size of the individual speaker nodes, while the edges between speakers represent the number of subsequent turns between two given speakers.

In the **Summary sentences** and **Keywords** components, the extracted important information is shown on two levels of granularity. The content in both components is ordered by time and is scrollable in the Entities component.

3.3.1 User interactions

All words or n-grams displayed in the Entities and Keywords component are clickable and change the state of the visualization. When a specific n-gram is selected, the Summary sentences component only displays sentences containing that word.

Figure 3.7: Visualization dashboard with the keyword "content" as a filter selection for the summary sentences.

In this way, a given user can control the specific information based on their interest in a particular topic. The next filter-based user interaction that can be done is selecting a specific time segment. This method allows the user to see summaries specific to a certain time.

With a timeframe selected, the components change their content to display the participants that spoke, the keywords and summary sentences, and the extracted entities specific to that time segment (figure 3.8).

Figure 3.8: Visualization dashboard with a specific time frame selected.

The **dimension** and **entity** aspects of the visualization should now be controllable by the user with these described methods. Additionally, the **granularity** of context is presented with three different granularity levels. The first level is the keywords, the second level is the summary sentences, and the last level is the summary sentence's context, which can be expanded by clicking on a specific sentence.

Figure 3.9: Selecting a specific sentence displays the full transcript text used to generate the sentence.

The methods used to process the raw data and prepare it for the visualization layer are presented in more detail in the next part.

Chapter 4

Models and algorithms

For simplicity, the backend layer is implemented as an API with different REST endpoints that serve the data to the visualization components. Each API endpoint has an underlying data processing and modeling technique described in this section and evaluated against different methods.

4.1 Entities

As a secondary component, the entity extraction uses a pre-trained transformer-based model. Specifically, a transformer-based model, trained on the Conll03 [Tjong et al., 2003] dataset, extracts the four categories: person, location, organization, and miscellaneous. In addition to providing extracted entities, the total number of each category is also presented in the visualization layer.

In the first evaluation in table 4.1, some of the standard models (Appendix A.1) for NER are compared on the conll03 test set. This test set contains 3453 word sequences and their corresponding NER-tags: {O, B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, B-MISC, I-MISC}. The B- tag describes a single entity or the beginning of a multi-word entity, while the I- tag describes words within a multi-word entity and always follows a B- tag. The 0 tag defines words, not as part of any entity.

	Precision	Recall	F1	Accuracy	Inference time
BERT large	0.909	0.922	0.915	0.983	46.48s
BERT	0.893	0.909	0.901	0.980	25.93s
DistilBERT	0.895	0.904	0.900	0.980	16.41s
Electra large	0.920	0.921	0.920	0.869	47.06s
RoBERTa large	0.921	0.935	0.927	0.870	49.12s

Table 4.1: Comparison of performance and runtime of different models on the Conll03 test dataset

The table shows that although most models perform similarly, the inference time can vary significantly between the fastest (DistilBERT) and the slowest model (RoBERTa), with is slightly below the best result of 94.6% recorded so far on the Conll03 dataset [Wang et al., 2020b]. As most meeting transcripts, especially from an ASR system, produce non-perfect data, another important factor that has to be considered is the performance of the individual models on uncased inputs. Since all entities are capitalized in English, recognizing them is significantly easier than with an all-lowercase input. The same dataset is transformed into lowercase input sequences to test this behavior and again fed to the models for inference. Table 4.2 shows the result of this evaluation.

	Precision	Recall	F1	Accuracy	Inference time
BERT large	0.734	0.278	0.403	0.878	48.16s
BERT	0.893	0.909	0.901	0.980	25.80s
DistilBERT	0.895	0.904	0.900	0.980	16.49s
Electra large	0.920	0.921	0.920	0.869	47.06s
RoBERTa large	0.908	0.799	0.850	0.865	49.41s

Table 4.2: Comparison of performance and runtime of different models on the Conll03 test dataset with lowercase input sequences

While the performance of most models is similar for a lowercase input, especially the large BERT model and the RoBERTa model, to some extent, show a decrease when running inference on uncased sequences.

As a final evaluation, a self-labeled dataset (Appendix A.1) is created from transcripts of short business meetings, each lasting between 5 and 15 minutes. The number of sequences in this dataset is 509, each with its corresponding NER tag with the categories described at the beginning of this section. This dataset should give more insights into how the selected model performs in a natural environment where the input is distorted and contains transcription errors.

	Precision	Recall	F1	Accuracy	Inference time
BERT large	0.613	0.711	0.658	0.985	6.66s
BERT	0.596	0.644	0.619	0.983	3.83s
DistilBERT	0.582	0.597	0.589	0.983	2.33s
Electra large	0.621	0.651	0.636	0.971	6.71s
RoBERTa large	0.712	0.731	0.721	0.973	7.14s

Table 4.3: Comparison of performance and runtime of different models on the self-labeled dataset for NER

Here, on this more impure dataset, the gap in performance between the different models is quite clearly visible. It also shows with RoBERTa that the model with the longest inference time also performs best.

While the computation time is essential for the implementation as part of the backend layer, different NLP techniques can be applied parallel within different workers, with results being cached. This is why for entity recognition, the model is picked which performs best in terms of F1 measure (RoBERTa) while taking the tradeoff of a longer inference time into account.

4.2 Sentiment and Speaker Change

Pretrained on the SST2 [Socher et al., 2013] dataset for binary classification, the sentiment data is produced with a sentiment classification model that outputs Positive (1) and Negative (0) tags at the sentence level. For a generation of a graph along the time axis, the individual sentiment tags $tag \in \{1, 0\}$ are aggregated per time frame:

$$score_t = \sum_{i=0}^{k} tag_i \tag{4.1}$$

where k is the number of sentiment tags within a given time segment.

The first evaluation in table 4.4 shows some selected sentiment classification models (Appendix A.2) that were evaluated on the SST2 test set. This test set contains 872 sentences together with their corresponding sentiment tag:

	Precision	Recall	F1	Accuracy	Inference time
DistilBERT	0.897	0.930	0.914	0.911	3.87s
RoBERTa	0.938	0.946	0.942	0.940	6.29s
BERT	0.796	0.818	0.807	0.800	6.18s
MultinomialNB	0.512	0.977	0.672	0.514	0.31s
LogRegression	0.549	0.760	0.637	0.560	0.34s

Table 4.4: Comparison of performance and runtime of different models on the SST2 dataset

Together with the pre-trained transformer architectures, the table shows two more traditional Machine Learning methods in Naive Bayes and Logistic Regression for comparison. While the inference on those methods is very time efficient, the performance decreases significantly from the transformer-based methods. Nevertheless, with 94% accuracy, the RoBERTa model achieves almost

state-of-the-art performance on the SST2 dataset, which is currently set at 97% with the Smart-RoBERTa model [Jiang et al., 2019].

In the next experiment, the same models are tested on a self-labeled dataset (Appendix A.2), created from transcripts of short business meetings, containing 509 sentences in total, together with their sentiment tag ($tag \in \{1, 0\}$):

	Precision	Recall	F1	Accuracy	Inference time
DistilBERT	0.883	0.696	0.778	0.705	2.44s
RoBERTa	0.875	0.789	0.830	0.739	3.75s
BERT	0.785	0.571	0.662	0.566	3.63s
MultinomialNB	0.748	0.926	0.827	0.713	0.12s
LogRegression	0.832	0.368	0.510	0.475	0.12s

Table 4.5: Comparison of performance and runtime of different models on the self-labeled dataset for Sentiment

Interestingly, the Naive Bayes implementation performs very well on this dataset and almost outperforms the RoBERTa transformer, only taking 0.12s for inference. One reason could be that the class distribution of the self-labeled dataset is different from the SST2 dataset. In the SST dataset, the positive classes represent 50.9% of the labels, while in the self-labeled dataset, they represent 74.2% which favors a model which tends to overpredict negative labels.

The speaker change graph is built purely from the transcript data, where a *speaker change* is defined as a switch from one participant to the next. All changes in speakers are counted and aggregated for a given time segment to represent the total number of changes for this given segment.

$$changes_t = \sum_{i=0}^{k} change_i \tag{4.2}$$

Where k is the number of changes within a given time interval, this metric should indicate which type of conversation (active or more monologic discussion) occurs during the segment.

4.3 Speaker Network

The nodes of the speaker network represent each participant in the meeting. The size is calculated utilizing the duration of each participant's utterances. For a given speaker, the size of the node is determined by:

$$size_i = \frac{\sum_{t=0}^{T} s_{it} - min(S)}{max(S) - min(S)} \tag{4.3}$$

S represents the sentence durations for all speakers and s_{it} a sentence for speaker i, resulting in a value between 0 and 1 for each participant's node. The edges between the speakers intend to indicate which participants are interacting with each other. The thickness of each edge is determined by how many subsequent turns two speakers have. For two given speakers, A and B, the edge weight is calculated as follows:

$$w(A, B) = \sum turn(A, B) + \sum turn(B, A) \tag{4.4}$$

where

$$turn(A, B) = \begin{cases} 1 & \text{if Speaker B follows Speaker A} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

4.4 Keywords and Summary sentences

The core parts of the backend layer are the keywords and summary sentences, as they can provide the most information to the user. One visualization layer functionality dictates that keywords must be tightly coupled with the summary sentences to apply a filter. To achieve this, the summary sentences generated drive the keywords, which are extracted using a custom algorithm.

4.4.1 Summary sentences

In the first step, a BART-based transformer model for summary extraction is trained on specific conversational datasets such as QMSUM [Zhong et al., 2021], SamSum [Gliwa et al., 2019] and general summary extraction datasets such as SemEval2017 [Augenstein et al., 2017], and MAKED [Verma et al., 2022]. The training was done for 34000 steps on a Tesla T4 GPU. Table A.5 shows the model parameters used.

For summary extraction, the self-trained model is compared with other commonly used models on the SamSum test set. This test set contains 819 source-text and summary pairs. The evaluation results can be seen in the following table 4.6

	Rouge1	Rouge2	RougeL	Inference time
BART finetuned	0.479	0.216	0.422	486s
BART large	0.403	0.203	0.312	496s
DistilBART	0.399	0.202	0.307	307s
T5	0.388	0.161	0.295	226s

Table 4.6: Comparison of performance and runtime of different models on the SamSum test set

As expected, the fine-tuned BART model outperforms the other, more general architectures while being competitive with the best-reported RougeL score on the SamSum dataset of 0.484 [Rohde et al., 2021]. Like in previous experiments, the performance of the selected models is also evaluated on a self-labeled dataset (A.3) for a better overview of what can be expected in a realistic setting.

	Rouge1	Rouge2	RougeL	Inference time
BART finetuned	0.489	0.255	0.391	9s
BART large	0.481	0.236	0.359	9s
DistilBART	0.459	0.241	0.347	6s
T5	0.396	0.185	0.305	3s

Table 4.7: Comparison of performance and runtime of different models on the self-labeled dataset for summary generation

While still showing significant results, one thing must be noted about the self-labeled dataset. In the set context of a meeting, transcripts are mainly generated with an ASR system which, to date, produces imperfect results depending on many factors such as the recording setting, speaker's language expertise, and model tuning. These introduced ASR errors were corrected in a manual post-processing step in the self-labeled dataset.

Intuitively, the input into the summarization model should influence the quality of the predictions. As an experiment to verify this case, the original audio recordings on which the self-labeled dataset is based are manually distorted and fed into different ASR systems, producing transcripts of different quality. The cosine similarity between each output and the corrected (gold) output is measured to see the ASR output's impact on the summary quality. Examples of different sentences with their respective similarity measure to the ground truth can be seen in table 4.4.1. This approach produces 14 transcript outputs with cosine similarity ranging from 0.55 to 0.95. Each output contains one full transcript containing 175 sentences.

Cosine similarity	ASR output
0.18	yes from oyesterday supporte trop with a d p obdimisation the did some more work on the concept of the bate trange
0.42	yes from my side, I supporte trop with a d p obdimisation the did some more work on the concept of the bate trange
0.75	yes from my side, I supported trop with a d p optimization the did some more work on the concept of the budget application
1.0 (gold)	Yes, from my side, yesterday I supported the DPV optimization, did some more work on the concept of the budget application

Table 4.8: Cosine similarities between different ASR outputs and the corrected gold output

In the next step, the finetuned model is used to produce summary sentences for each transcript version to evaluate the effect of the input on the produced summaries. Each summary is evaluated in terms of RougeL performance against the correct summary. In Figure 4.1, the cosine similarity between the ground truth (gold) and each transcript is plotted against the RougeL of the produced summary. A linear regression is added to the visualization to check for possible relationships. The listed data points of the plot are in table A.4.

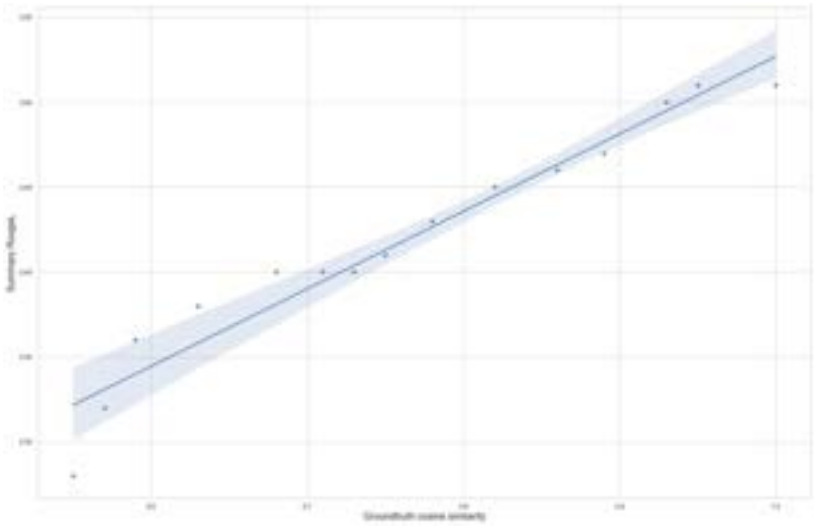

Figure 4.1: Datapoints with the cosine similarities of different ASR outputs against their produced summary RougeL metric.

This figure shows an obvious linear relationship between the quality of the input and the quality of the produced summary with the regression equation of:

$$Y = 0.41X + 0.11 \tag{4.6}$$

Another interesting finding is that the summary is still grammatically reasonable, even with poorly transcribed source texts. The following two example sentences show this behavior. When comparing the two sentences, the first summary has some correct information but includes semantical errors.

Cosine similarity	**Generated summaries**
0.55	Ivan has almost finished working on a conting page and is going to move it staging. Philipp is **looking forward to the meeting** and will take a look at the page if it works.
0.89	Ivan will move the content page to staging. The other people will keep an eye on staging and will text Ivan if they find anything wrong with the navigation.

Table 4.9: Generated summaries with different input quality

This is an important finding in a practical setting: summaries might be misleading by including false information if the source text is insufficient.

Additionally, when using the summaries within the visualization layer, the meeting transcript is split into fixed-length segments and fed into the trained summarization model. For each segment, between one to five summary sentences are generated. The amount depends on the content and information within the source segment.

4.4.2 Keywords

After creating summary sentences, the keywords are generated based on the source text and the generated summary sentences. The underlying assumption is that any n-gram that overlaps within the source text and its condensed summary is crucial and must be identified as a keyword. Additionally, it is worth noting that these keywords serve as a filtering mechanism in the designed application, and therefore they must align with the generated summary sentences to be effective.

The algorithm used for extraction is two-part. The first part creates a hashtable out of the tokenized source text. In this hashtable, the keys are formed from individual words, with the values being sets of their successive words (Algorithm 1).

Algorithm 1 Source text to hashtable conversion

```
list ← tokenize(source_text)
seen ← set()
word_links ← hashtable
for i in list do
    if i < len(list) − 1 then
        next_word ← list[i + 1]
    else
        next_word ← None
    end if
    if list[i] in seen then
        push next_word to word_links[list[i]]
    else
        word_links[list[i]] ← set([next_word])
    end if
    seen[list[i]] ← list[i]
end for
```

Algorithm 2 Keyword matching algorithm

```
word_links ← hashtable
list ← tokenize(summary)
max_ngram_size ← int
visited_idxs ← set()
found_keywords ← []
for i in list do
    if i not in visited_idxs and list[i] in word_links then
        keyword_ngram ← [list[i]]
        push i to visited_idxs
        j = 1
        is_match ← True
        while j < max_ngram_size and is_match do
            if len(list) > i + j and list[i + j] in word_links[list[i + j − 1]] then
                push list[i + j] to keyword_ngram
                push [i + j] to visited_idxs
                j ← j + 1
            else
                is_match ← False
            end if
        end while
        push keyword_ngram to found_keywords
    end if
end for
```

In the next step, the algorithm looks up each word of the tokenized summary

text in the created source text hashtable. If the hashtable contains the word, the following word at index position $i + 1$ gets checked for its existence in the value set of the hashtable key. If it exists as the next word in both texts, an n-gram of size two is created, and the successive word at $i + 1$ is used as a new hashtable key for lookup. This process is repeated up to the maximum n-gram size for matching subsequences (Algorithm 2).

An example that shows this matches visually can be seen in 4.2. After tokenization and stop-word removal, the source text is presented on the y axis, while the tokenized summary sentence is seen on the x axis. The blue marks show matches between the source text and its summary, while fields that are diagonal adjacent represent keywords consisting of multiple words.

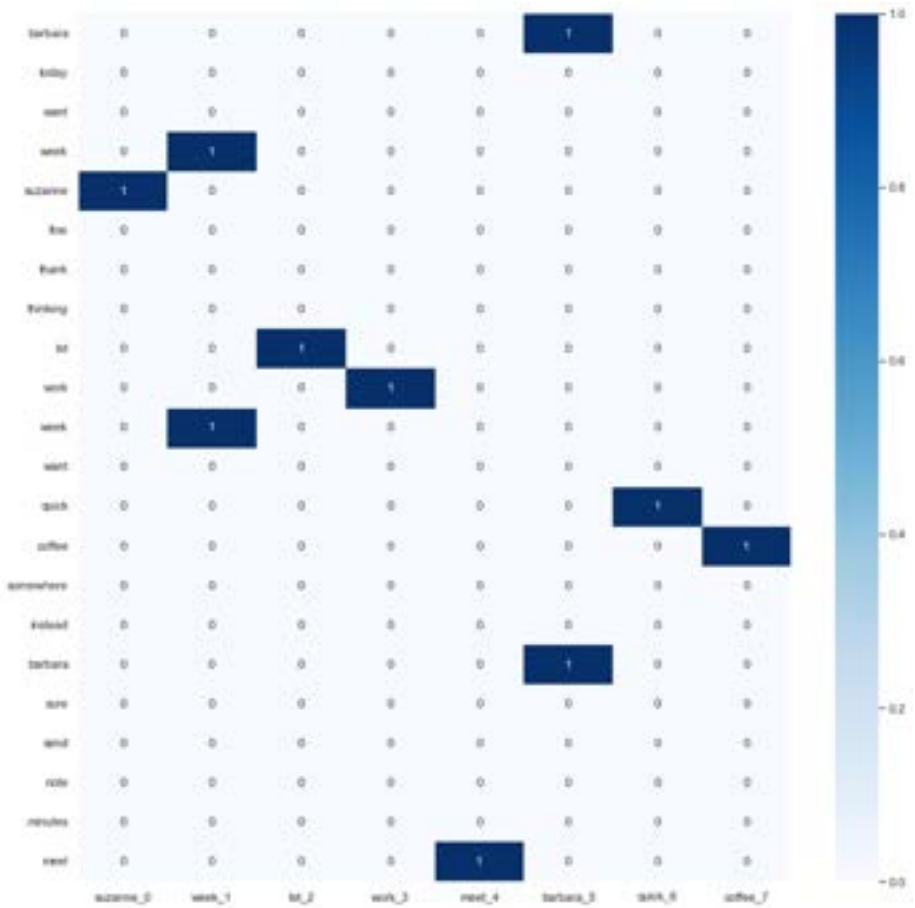

Figure 4.2: Matching words between the source text and its summary. Diagonal adjacent matches form multi-word keywords.

For the example above, the formed keywords would be "lot work" and "quick coffee," which can already give a good intuition about the contents of this sentence. Together with the summary sentences, the keywords provide a quick way

to get an overview and filter only the information needed as described in 3.3.1.

With the visualization layer and the underlying models in place, the next part analyzes all the system components with a user test, evaluating the built methods on different scenarios and questions.

Chapter 5

User test

5.1 Setup

In this chapter, a qualitative user study is designed and conducted to evaluate the effectiveness of the proposed interface, with the primary goal of understanding the essential information of discussions within a short time. Within this study, the usefulness and usability of the proposed interface are compared with those of a baseline system.

A 50-minute meeting transcript from the AMI corpus is used for the study. In the transcribed meeting, the four participants are having a product meeting about the functional design of a new remote control. The participants have project manager, designer, marketing, or user interface roles. They introduce new requirements and present their research in each category to the group, such as customer needs, target group, and pricing.

The baseline system is provided as a formatted word document with a speaker and time information, with standard tools such as text search enabled. Each test user is either assigned to answer the questions using the designed application or provides the baseline by answering the questions utilizing purely the transcript.

The user test is conducted on the LimeSurvey [1] platform between the 09. January and the 15. February 2023, with most participants being students or work colleagues. The users that participate via the designed application are suggested to view a two-minute instruction on the general usage of the tool and how to interact with it to find the necessary information. The study task is designed to be solved within a reasonable time frame of 15-25 minutes.

5.2 Task design

Within the task, eleven questions about the meeting are to be answered. Five have the multiple-choice, categorical answer, while the other six allow for a free-

[1] https://survey.webcenter.ch/limesurvey2018

text response. The response and the time needed to answer the question are stored for each question. Questions cover different aspects of the meeting and contain some general/overall questions and topic-specific questions.

General / Overview

One set of questions is more general. It aims to evaluate the ease and efficiency of the tool allowing the user to access high-level information about the meeting. Considering this should help determine whether the summarization tool is effective for quickly and easily grasping the overall purpose and content of the meeting. Some of the questions asked are:

- Who was present at the meeting?
- What was the agenda of the meeting?
- Who led the discussion?

Topic-specific

The second set of questions is more topic-specific and asks the user about the specifics of a particular topic covered in the meeting. Again, this should help determine whether the tool can capture all the essential information. Those questions ask for more context with examples such as:

- What age is the target customer group?
- Why was this target group picked?
- What are the customer needs?

The questions are designed to cover a range of difficulty levels to ensure that the task is suitable for participants with varying levels of familiarity with the meeting and their affinity to online tools, ensuring that all participants find the assignment challenging but straightforward enough. After the user test, an additional question is appended to the task, prompting participants to provide optional feedback regarding their likes and dislikes about the experience with the meeting summarization tool. This question aims to identify areas for improvement that may have yet to be captured through the structured set of questions.

With the user test designed, the next step is to evaluate the tool's effectiveness compared to the baseline transcript. In this section, the data collected from the user test is analyzed, and the results are discussed, providing insights into the strengths and weaknesses of the tool and potentially identifying areas for improvement.

5.3 Evaluation

The first evaluation of the user test compares the number of participants that took part in the test via the designed application, compared with the number of participants that took part in the user test using purely the transcript as the aid method.

The raw data collected from the user test revealed that the number of entries with at least one answer filled out was 72 for the application and 25 for the transcript, shown in figure 5.1. However, a preliminary filtering step was performed to ensure an unbiased evaluation of the results, particularly when analyzing the time to complete each answer. This step involves retaining only those entries that exceed a 30% threshold for correct answers, accounting for the possibility of incomplete or inaccurate responses that may have skewed the results.

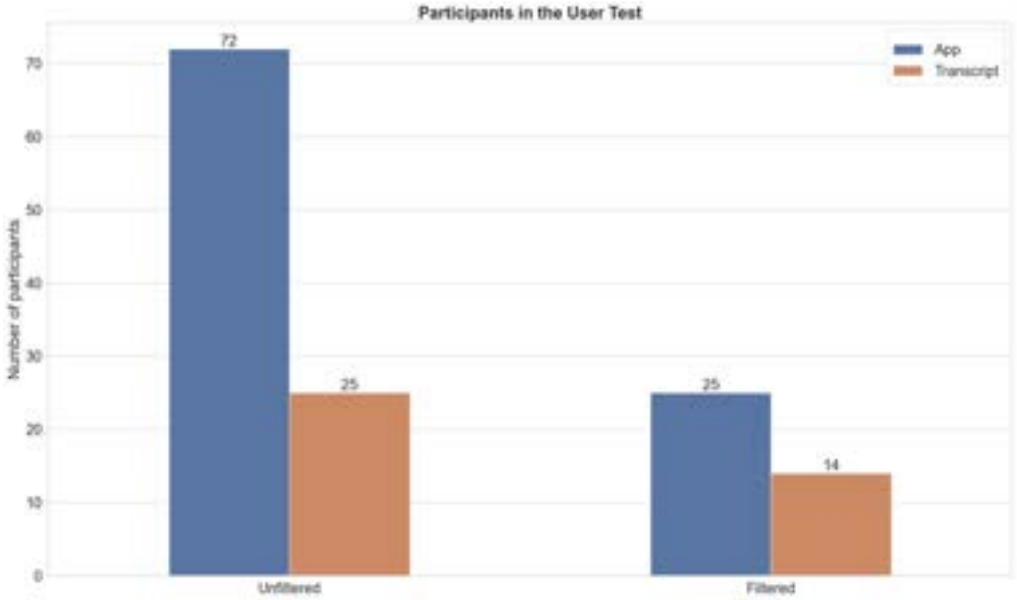

Figure 5.1: Number of participants in the User Test.

In detail, the filtering process is accomplished by comparing every answer submitted in the test to a predefined set of reference words that serve as the benchmark or ground truth. In the case of multiple-choice answers, an exact match is considered valid, while a match with at least one of the reference keywords is deemed sufficient for questions requiring free-text responses.

An example is provided in table 5.1 for one of the questions, **"Why was this target group picked?"** which concerns the discussion of the target group the team is targeting with their remote control. The correct reference keywords for this question, as mentioned in the transcript, include: "expendable income," "use technology," "young professionals," "computer daily," and "money to spare."

Answer	Valid
"...**young professionals** that have **expendable income**"	Yes
"...willing to **use technology**"	Yes
"...they use a **computer daily**"	Yes
"...it's the older generation "	No

Table 5.1: Initial filtering of the answers. Answers with matching keywords to the reference are considered valid.

To pass the initial filtering step, at least 30% of the responses provided by a participant must be valid when applying this approach to all questions, leading to a total number of 25 answers for the app, respectively 14 for the transcript after filtering.

5.3.1 Correctness of the responses

As the next step, the overall quality of the responses is evaluated, meaning the correctness of the answers.

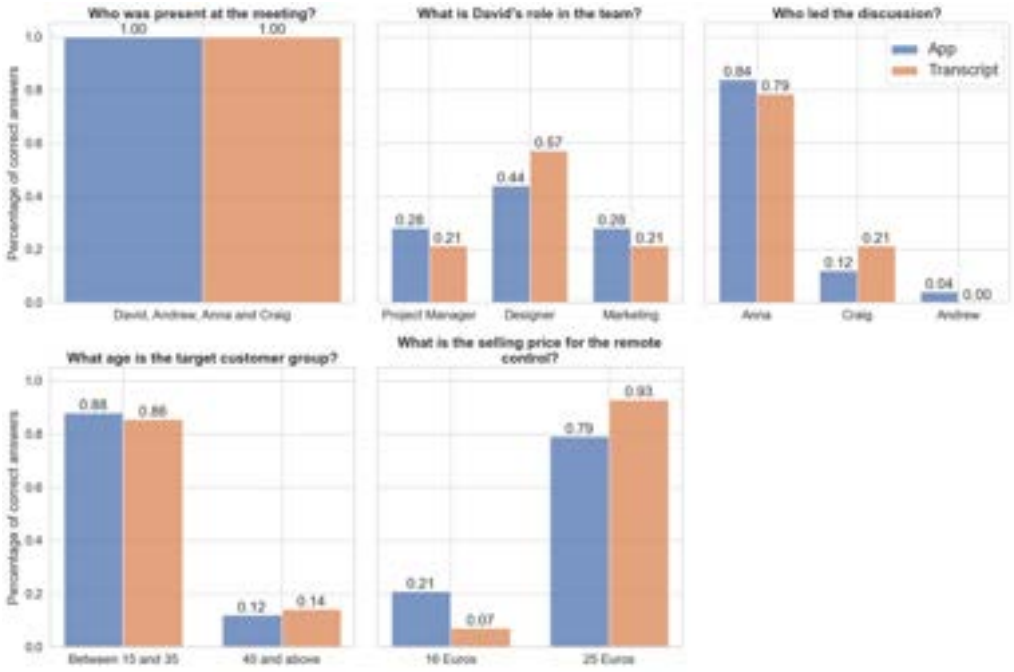

Figure 5.2: Percentage of chosen categorical answers in the user test, comparing the application with the answers from utilizing merely the transcript.

The questions are split according to their response type (multiple-choice or free-text) to achieve this. Figure 5.2 shows the results for the questions with multiple-choice, categorical answers. The score for comparison is shown in percentages to adjust for the reduced number of responses in the transcript version of the user test.

It should be noted that the initial question served as a "concentration test" for the participants, meaning that only those who provided correct solutions were permitted to proceed with the test, resulting in a 100% score for this question in both categories. The initial rationale behind this decision was to get more qualitative responses and reduce the likelihood of incomplete or erroneous responses, improving the accuracy and reliability of the findings.

The presented figure indicates that the responses across all categories are primarily comparable concerning their accuracy. The most significant discrepancy observed is a modest 14% deviation in the question related to the selling price of the remote control. Notably, in each of the questions posed, the category with the most responses aligns with the expected correct answer.

A rating system analogous to the initial filtering process is implemented to assess the accuracy of the free-text responses which fall within the other category. For each question, the answer is evaluated to determine if it contains any of the designated target words. If a target word is present, a score of 1 is assigned; otherwise, a score of 0 is recorded. The results of this scoring can be seen in figure 5.3.

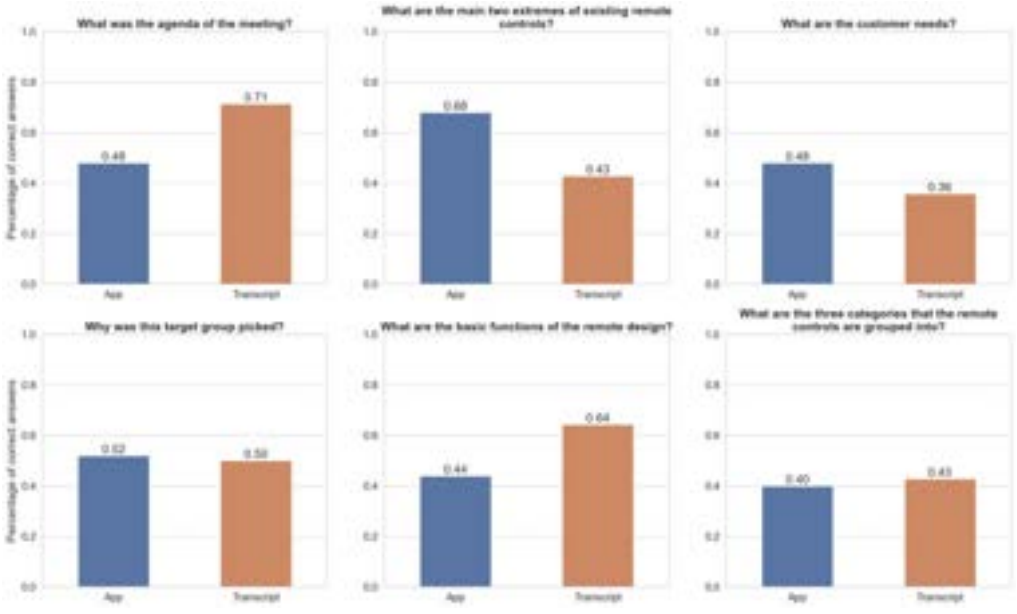

Figure 5.3: Percentage of chosen free-text answers in the user test, comparing the application with the answers from utilizing merely the transcript.

Once again, the scores attained by users responding to the questions via the app and the transcript are comparable. The most significant deviation observed is 23% in favor of the transcript users for the question "What was the agenda of the meeting?" and 25% for users utilizing the application when responding to the question "What are the main two extremes of existing remote controls?" respectively.

The following figure 5.4 provides an overarching comparison of all questions, contrasting those featuring categorical, multiple-choice answers with those containing free-text responses.

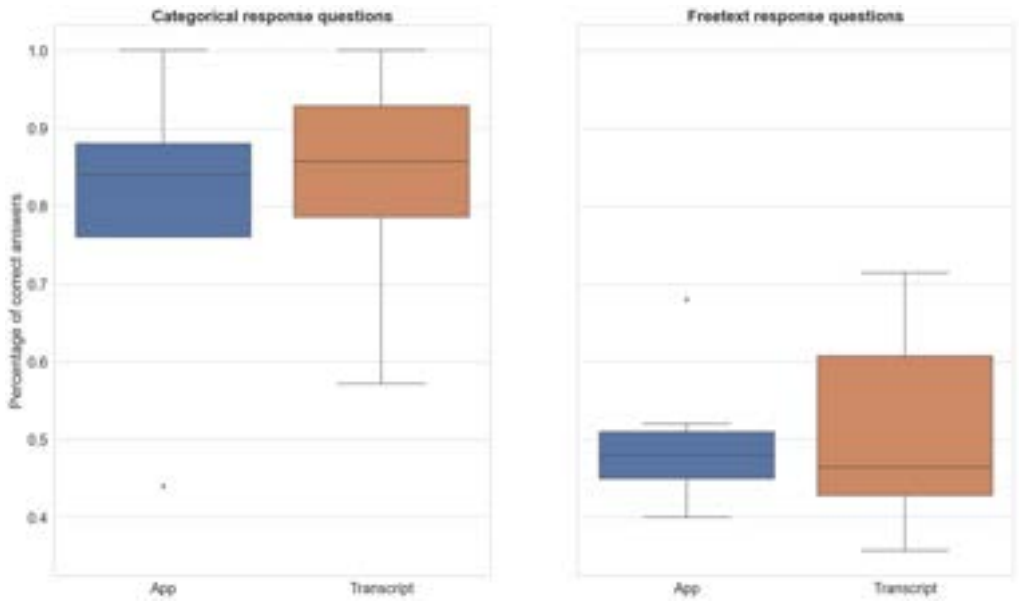

Figure 5.4: Overall comparison of the correctness of the responses between the categorical and free-text responses and the application vs. transcript.

5.3.2 Response times

In addition to evaluating the correctness of participants' responses, the speed at which they provided their answers is also measured as part of the user test. For each question, the time taken to answer was recorded, and this information was used to compare the response times between the application and the transcript.

It is important to note that only the response times of correct answers were considered in the time analysis. Figure 5.5 and table 5.3 present this analysis's results, comparing the time distribution needed to answer each question in seconds.

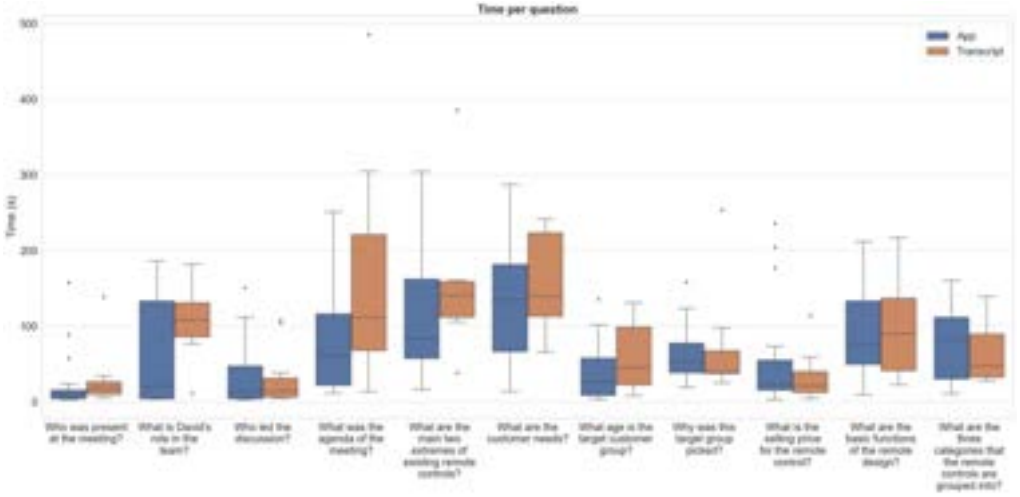

Figure 5.5: The time distribution on the participants' speed for answering the individual questions, comparing the application users with those using the transcript.

Question	Mean(App)	Mean(Transcript)	Delta
Question 1	19.49s	25.75s	-6.26s
Question 2	65.97s	107.56s	-41.59s
Question 3	34.08s	31.13s	2.95s
Question 4	84.03s	155.82s	-71.79s
Question 5	107.03s	161.94s	-54.91s
Question 6	132.56s	156.53s	-23.97s
Question 7	39.30s	59.03s	-19.73s
Question 8	66.70s	73.12s	-6.42s
Question 9	48.06s	33.20s	14.86s
Question 10	95.93s	97.26s	-1.33s
Question 11	73.83s	65.44s	8.39s

Table 5.2: Table of average response times of each question.

An interesting finding emerged from the analysis of the response time data. Specifically, questions that took longer to answer - potentially indicating increased difficulty or a need for contextual understanding - were responded to more quickly by participants using the application than those using the transcript. In contrast, questions that required participants to locate specific information mentioned during the meeting, such as question 9 regarding the selling price of the remote control, led to generally faster response times in the transcript.

The next figure 5.6 presents the overall time comparison between the two user groups.

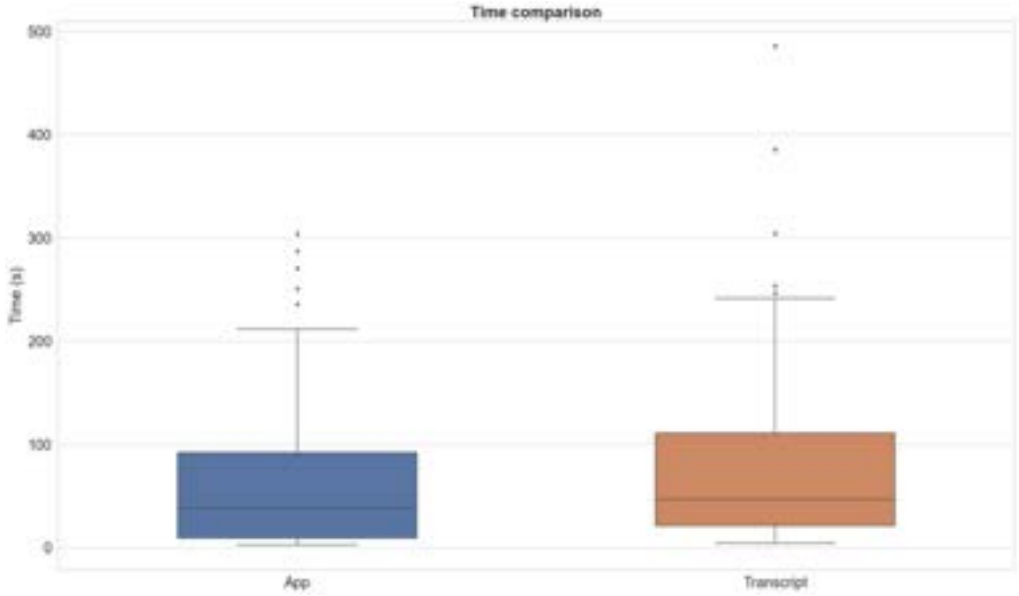

Figure 5.6: The time distribution on the participants' speed for answering the individual questions, summarized for all questions.

Mean(App)	Mean(Transcript)	Delta
63.93s	78.89s	-14.96s

Table 5.3: Table of average response times, summarized for all questions.

The table shows that there is indeed an overall difference in response times between participants who used the self-developed application and those who used the transcript to answer the questions, which is 15 seconds when comparing both averages.

5.3.3 User feedback

In addition to the questions regarding the meeting topic, users were also asked to provide feedback on their experience using the application. When asked what they liked about the app and what could be improved, the participants proposed various recommendations.

Several participants gave positive feedback regarding the usefulness of the summary sentences provided by the application. However, some suggested grouping or tagging by topic to make them more manageable.

Another common topic in the feedback was the need for a search function within the application. The suggestion was that a search function could be used to locate better specific keywords and the extended possibility to search for synonyms. Other individual feedback responses included incorporating the audio from the meeting, that the application could benefit from fewer buttons, and general technical issues with the application, including slow loading times and instability.

Concluding the evaluation, the findings indicate that both user groups had the same score regarding the correctness of the answers. However, regarding the response time needed to answer, users with access to the application were significantly faster, especially on questions requiring more contextual understanding. In contrast, users who only had access to the transcript performed better on some questions that required locating specific information mentioned during the meeting.

The findings suggest that a supporting application can significantly benefit a faster grasp of context and more complex information during a meeting. However, using transcripts can still be valuable for searching and pinpointing specific information and should be integrated into the supporting application to provide a more comprehensive user experience.

It is important to note that the sample size of this study was limited and intended to be qualitative. Therefore, further research with a larger sample size must verify these findings and provide a more accurate representation of the user experience. Additionally, the study did not investigate the potential impact of user familiarity with technology, which could affect the performance of the two user groups.

Chapter 6

Conclusion

This thesis shows the design and implementation steps of building a visual summarization system for meetings. With many moving parts, designing such a system to address the goal of understanding essential information interactively and efficiently concerning time is challenging and requires many iterations.

Chapter 1 motivates the underlying topic and outlines the general application scenario and supporting sub-goals for the main objective. A background about the key technologies within text and dialogue summarization and their usage in visualization applications is presented in chapter 2. This chapter also provides the relevant preliminaries of the NLP techniques that drive the design and experiments of this thesis. The design process of the visual application layer is presented in chapter 3, with the underlying models and algorithms needed to serve this application introduced in chapter 4. Chapter 5 shows the setup, layout, and evaluation of the user test used to assess the designed application's effectiveness to compare the correctness and time of the user to respond to each question in the application with users using merely the transcript.

The following subsections analyze the extent to which the goals of the thesis have been achieved and explore potential future work that can be done to improve and build upon the proposed system.

6.0.1 Results

The proposed system incorporates different NLP techniques, such as entity recognition, sentiment analysis, and summarization, and a self-designed keyword extraction method serving as a filtering mechanism for the summary sentences within the visualization layer.

The conducted user test on this built application, while not on a large scale, shows a significant reduction in time needed to get to know details about a topic and be able to answer questions about it. Specifically, the application users demonstrated an average response time of 15 seconds shorter than those who used only the transcript, with individual questions having a reduced average response

time by over one minute.

This observed response time reduction indicates that the system has successfully achieved the intended goals of concisely summarizing a given meeting and facilitating user interaction with their topic of interest.

6.0.2 Future Work

Several areas require further investigation and improvement. For example, one potential avenue for the modeling and algorithmic part is to explore the use of additional advanced NLP techniques, such as query-based summarization, which could give the user even more flexibility to interact with the data by providing specific topics of interest that the summary should include.

Another improvement to this system could be the inclusion of the audio file, allowing further analyzing of the meeting for the user and also extending the application's capabilities to highlight important information in the discussion by examining the participants' speech and detecting changes in pitch and change of voice.

Moreover, as also mentioned by participants of the user test, the interaction in the application, including its stability and individual loading times, is an area of improvement. Designing an application is an iterative process. It would help to conduct a more detailed user study and evaluate what features the users use and what is missing (such as a keyword search function in the proposed application) to provide an even better experience.

Bibliography

[Ahmad et al., 2018] Ahmad, M., Aftab, S., Salman, M., and Hameed, N. (2018). Sentiment analysis using svm: A systematic literature review. *International Journal of Advanced Computer Science and Applications*, 9.

[Allahyari et al., 2017] Allahyari, M., Pouriyeh, S. A., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. J. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *CoRR*, abs/1707.02919.

[An et al., 2021] An, C., Zhong, M., Chen, Y., Wang, D., Qiu, X., and Huang, X. (2021). Enhancing scientific papers summarization with citation graph.

[Asahara and Matsumoto, 2003] Asahara, M. and Matsumoto, Y. (2003). Japanese named entity extraction with redundant morphological analysis.

[Augenstein et al., 2017] Augenstein, I., Das, M., Riedel, S., Vikraman, L., and McCallum, A. (2017). SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

[Baccianella et al., 2010] Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. volume 10.

[Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.

[Baumel et al., 2018] Baumel, T., Eyal, M., and Elhadad, M. (2018). Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models.

[Campos et al., 2018] Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A. (2018). Yake! collection-independent automatic keyword extractor.

[Cao et al., 2012] Cao, N., Lin, Y.-R., Sun, X., Lazer, D., Liu, S., and Qu, H. (2012). Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Transactions on Visualization and Computer Graphics*.

[Chopra et al., 2016] Chopra, S., Auli, M., and Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California. Association for Computational Linguistics.

[Chou and Yang, 2011] Chou, J.-K. and Yang, C.-k. (2011). Papervis: Literature review made easy. *Comput. Graph. Forum*, 30:721–730.

[Cohan et al., 2018] Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Dey et al., 2016] Dey, L., Chakraborty, S., Biswas, A., Bose, B., and Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and K-NN classifier. *CoRR*, abs/1610.09982.

[Dusart et al., 2021] Dusart, A., Pinel-Sauvagnat, K., and Hubert, G. (2021). Tssubert: Tweet stream summarization using bert.

[Dörk et al., 2010] Dörk, M., Gruen, D., Williamson, C., and Carpendale, S. (2010). A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, pages 1129–1138.

[Gliwa et al., 2019] Gliwa, B., Mochol, I., Biesek, M., and Wawer, A. (2019). SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

[Goodfellow et al., 2015] Goodfellow, I. J., Bengio, Y., and Courville, A. C. (2015). Deep learning. *Nature*, 521:436–444.

[Hulth, 2003] Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.

[Humayoun et al., 2017] Humayoun, S. R., Ardalan, S., AlTarawneh, R., and Ebert, A. (2017). TExVis: An Interactive Visual Tool to Explore Twitter Data. In Kozlikova, B., Schreck, T., and Wischgoll, T., editors, *EuroVis 2017 - Short Papers*. The Eurographics Association.

[Jacobs and Rau, 1993] Jacobs, P. S. and Rau, L. F. (1993). Innovations in text interpretation. *Artificial Intelligence*, 63(1):143–191.

[Jia et al., 2020] Jia, R., Cao, Y., Tang, H., Fang, F., Cao, C., and Wang, S. (2020). Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3622–3631, Online. Association for Computational Linguistics.

[Jiang et al., 2019] Jiang, H., He, P., Chen, W., Liu, X., Gao, J., and Zhao, T. (2019). SMART: robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *CoRR*, abs/1911.03437.

[Kaur et al., 2018] Kaur, S., Sikka, G., and Awasthi, L. K. (2018). Sentiment analysis approach based on n-gram and knn classifier. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pages 1–4.

[Koay et al., 2020] Koay, J. J., Roustai, A., Dai, X., Burns, D., Kerrigan, A., and Liu, F. (2020). How domain terminology affects meeting summarization performance.

[Koay et al., 2021] Koay, J. J., Roustai, A., Dai, X., and Liu, F. (2021). A sliding-window approach to automatic creation of meeting minutes.

[Kryscinski et al., 2019] Kryscinski, W., McCann, B., Xiong, C., and Socher, R. (2019). Evaluating the factual consistency of abstractive text summarization. *CoRR*, abs/1910.12840.

[Lee et al., 2009] Lee, Y.-J., Bae, M.-J., Woo, G., and Cho, H.-G. (2009). A personalized visualizing and filtering system for a large set of responding messages on internet discussion forums. In *2009 Ninth IEEE International Conference on Computer and Information Technology*.

[Lewis et al., 2020] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

[Li and Zhang, 2021] Li, Q. and Zhang, Q. (2021). Twitter event summarization by exploiting semantic terms and graph network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):15347–15354.

[Lin, 2004] Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

[Liu, 2010] Liu, B. (2010). Sentiment analysis and subjectivity.

[Liu et al., 2015] Liu, S., Chen, Y., Wei, H., Yang, J., Zhou, K., and Drucker, S. M. (2015). Exploring topical lead-lag across corpora. *IEEE Transactions on Knowledge and Data Engineering*, 27(1):115–129.

[Liu et al., 2014] Liu, S., Wang, X., Liu, J., Chen, J., Zhu, J., and Guo, B. (2014). Topicpanorama: a full picture of relevant topics. In *IEEE Conference on Visual Analytics Science and Technology (IEEE VAST)*. IEEE.

[Liu et al., 2009a] Liu, S., Zhou, M. X., Pan, S., Qian, W., Cai, W., and Lian, X. (2009a). Interactive, topic-based visual text summarization and analysis. page 543–552.

[Liu et al., 2009b] Liu, Z., Li, P., Zheng, Y., and Sun, M. (2009b). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266, Singapore. Association for Computational Linguistics.

[Livestats, 2022] Livestats (2022). Livestats: Daily tweet estimation.

[Ma et al., 2020] Ma, C., Zhang, W. E., Guo, M., Wang, H., and Sheng, Q. Z. (2020). Multi-document summarization via deep learning techniques: A survey.

[Mihalcea and Tarau, 2004] Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

[Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

[Nakayama, 2018] Nakayama, H. (2018). seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

[Nandwani and Verma, 2021] Nandwani, P. and Verma, R. (2021). A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11.

[Nema et al., 2017] Nema, P., Khapra, M., Laha, A., and Ravindran, B. (2017). Diversity driven attention model for query-based abstractive summarization.

[Nenkova et al., 2011] Nenkova, A., Maskey, S., and Liu, Y. (2011). Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, page 3, Portland, Oregon. Association for Computational Linguistics.

[Nikolov, 2020] Nikolov, N. I. (2020). *Abstractive Document Summarization in High and Low Resource Settings*. PhD thesis, ETH Zurich, Zurich.

[Odumuyiwa and Osisiogu, 2019] Odumuyiwa, V. and Osisiogu, U. (2019). A systematic review on hidden markov models for sentiment analysis. In *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, pages 1–7.

[Oelke, 2010] Oelke, D. (2010). Visual document analysis: Towards a semantic analysis of large document collections.

[Papineni et al., 2002] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

[Raffel et al., 2019] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

[Ramesh et al., 2022] Ramesh, G., Manyam, V., Mandula, V., Myana, P., Macha, S., and Reddy, S. (2022). *Abstractive Text Summarization Using T5 Architecture*, pages 535–543.

[Repustate, 2022] Repustate (2022). Repustate: Customer and employee sentiment .

[Rohde et al., 2021] Rohde, T., Wu, X., and Liu, Y. (2021). Hierarchical learning for generation with long source sequences.

[Sanh et al., 2019] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

[Shang et al., 2018] Shang, G., Ding, W., Zhang, Z., Tixier, A. J.-P., Meladianos, P., Vazirgiannis, M., and Lorré, J.-P. (2018). Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization.

[Singh, 2020] Singh, M. (2020). Techcrunch: Daily whatsapp estimation.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

[Sonawane et al., 2019] Sonawane, S., Kulkarni, P., Deshpande, C., and Athawale, B. (2019). Extractive summarization using semigraph (essg). *Evolving Systems*, 10.

[Statista, 2022] Statista (2022). Statista: Daily email estimation.

[Talkwalker, 2022] Talkwalker (2022). Talkwalker: Consumer intelligence .

[Tixier et al., 2016] Tixier, A., Skianis, K., and Vazirgiannis, M. (2016). GoWvis: A web application for graph-of-words-based text visualization and summarization. In *Proceedings of ACL-2016 System Demonstrations*, pages 151–156, Berlin, Germany. Association for Computational Linguistics.

[Tjong et al., 2003] Tjong, S., Erik, F., and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

[Tomokiyo and Hurst, 2003] Tomokiyo, T. and Hurst, M. (2003). A language model approach to keyphrase extraction.

[Tunstall et al., 2022] Tunstall, L., von Werra, L., and Wolf, T. (2022). *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, Incorporated.

[Turpin et al., 2007] Turpin, A., Tsegay, Y., Hawking, D., and Williams, H. E. (2007). Fast generation of result snippets in web search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 127–134, New York, NY, USA. Association for Computing Machinery.

[Vaswani et al., 2017a] Vaswani, A., Shazeer, N., Parmar, N., and Polosukhin, I. (2017a). Attention is all you need. *CoRR*, abs/1706.03762.

[Vaswani et al., 2017b] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017b).

[Veličković et al., 2017] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2017). Graph attention networks.

[Verma et al., 2022] Verma, Y., Jangra, A., Saha, S., Jatowt, A., and Roy, D. (2022). MAKED: Multi-lingual automatic keyword extraction dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6170–6179, Marseille, France. European Language Resources Association.

[Visix, 2022] Visix (2022). Visix: Daily meetings estimation.

[Vontage, 2022] Vontage (2022). Vontage: Conversation analyzer.

[Wang et al., 2018] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

[Wang et al., 2020a] Wang, D., Liu, P., Zheng, Y., Qiu, X., and Huang, X. (2020a). Heterogeneous graph neural networks for extractive document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6209–6219, Online. Association for Computational Linguistics.

[Wang et al., 2019] Wang, D., Liu, P., Zhong, M., Fu, J., Qiu, X., and Huang, X. (2019). Exploring domain shift in extractive text summarization.

[Wang et al., 2020b] Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, Z., Huang, F., and Tu, K. (2020b). Automated concatenation of embeddings for structured prediction. *CoRR*, abs/2010.05006.

[Wu et al., 2021] Wu, C.-S., Liu, L., Liu, W., Stenetorp, P., and Xiong, C. (2021). Controllable abstractive dialogue summarization with sketch supervision.

[Xu and Durrett, 2019] Xu, J. and Durrett, G. (2019). Neural extractive text summarization with syntactic compression.

[Yan et al., 2019] Yan, H., Deng, B., Li, X., and Qiu, X. (2019). Tener: Adapting transformer encoder for named entity recognition.

[Zhang et al., 2021] Zhang, S., Celikyilmaz, A., Gao, J., and Bansal, M. (2021). Emailsum: Abstractive email thread summarization.

[Zhang et al., 2020] Zhang, Y., Tuo, M., Yin, Q., Qi, L., Wang, X., and Liu, T. (2020). Keywords extraction with deep neural network model. *Neurocomputing*, 383:113–121.

[Zhong et al., 2019] Zhong, M., Liu, P., Wang, D., Qiu, X., and Huang, X. (2019). Searching for effective neural extractive summarization: What works and what's next.

[Zhong et al., 2021] Zhong, M., Yin, D., Yu, T., Zaidi, A., Mutuma, M., Jha, R., Awadallah, A. H., Celikyilmaz, A., Liu, Y., Qiu, X., and Radev, D. (2021). Qmsum: A new benchmark for query-based multi-domain meeting summarization.

[Zhu et al., 2020] Zhu, C., Xu, R., Zeng, M., and Huang, X. (2020). A hierarchical network for abstractive meeting summarization with cross-domain pretraining.

[Zhu et al., 2019] Zhu, H., Dong, L., Wei, F., Qin, B., and Liu, T. (2019). Transforming wikipedia into augmented data for query-focused summarization.

Appendix A

Models and datasets

A.1 Named Entity Recognition

The following table shows the links for the used models for Named Entity Recognition:

Parameter	Value
BERT large	https://huggingface.co/51la5/bert-large-NER
DistilBERT	https://huggingface.co/51la5/distilbert-base-NER
Electra large	https://huggingface.co/51la5/electra-large-NER
RoBERTa large	https://huggingface.co/51la5/roberta-large-NER

Table A.1: Models used for named entity recognition

The self-labeled dataset can be found at https://huggingface.co/datasets/51la5/standups_ner.

A.2 Sentiment analysis

The following table shows the links for the used models for Sentiment analysis:

Parameter	Value
DistilBERT	https://huggingface.co/51la5/distilbert-base-sentiment
RoBERTa	https://huggingface.co/51la5/roberta-base-sentiment
BERT	https://huggingface.co/51la5/bert-base-sentiment
MultinomialNB	https://huggingface.co/51la5/MultinomialNB-sentiment
LogRegression	https://huggingface.co/51la5/LogRegression-sentiment

Table A.2: Models used for sentiment analysis

The self-labeled dataset can be found at https://huggingface.co/datasets/51la5/standups_sentiment.

A.3 Summary sentences

The following table shows the links for the used models for summary generation:

Parameter	Value
BART finetuned	https://huggingface.co/51la5/BART-QMSUM-Summary
BART large	https://huggingface.co/51la5/BART-large-summary
DistilBART	https://huggingface.co/51la5/distilBART-summary
T5 small	https://huggingface.co/51la5/T5-summary

Table A.3: Models used for summary generation

The self-labeled dataset can be found at https://huggingface.co/datasets/51la5/keyphrase-extraction.

A.4 ASR evaluation

This table shows the evaluation results with ASR transcripts of different qualities. The cosine similarity is measured between the corrected (gold) transcript and the respective ASR output before being fed into the summarization model.

ASR output cosine similarity	Summary RougeL
0.328287	0.281136
0.339373	0.320540
0.353738	0.358766
0.385754	0.383107
0.418324	0.398125
0.434445	0.402000
0.441651	0.404016
0.455698	0.413752
0.471639	0.426243
0.497084	0.454340
0.524322	0.456569
0.547836	0.467236
0.564079	0.498470
0.573044	0.507604

Table A.4: Cosine similarity impact of summarization performance

Parameter	Value
min length	56
max length	142
no repeat ngram size	3
num beams	4
num hidden layers	12
vocab size	50264

Table A.5: Summarization model config parameters

Appendix B

User Test results

This section shows the questions and answers of the user test.

B.0.1 Questions

Question	Text
Question 1	Who was present at the meeting?
Question 2	What is David's role in the team?
Question 3	Who led the discussion?
Question 4	What was the agenda of the meeting?
Question 5	What are the main two extremes of existing remote controls?
Question 6	What are the customer needs?
Question 7	What age is the target customer group?
Question 8	Why was this target group picked?
Question 9	What is the selling price for the remote control?
Question 10	What are the basic functions of the remote design?
Question 11	What are the three categories that the remote controls are grouped into?

Table B.1: User test questions

B.0.2 Application users

The following tables show the answers of the users utilizing the application for the user test.

Id	Submit date	Lastpage	Seed	Question 1	Question 2	Question 3	Question 4
1	2023-01-13 8:48:20	12	859783367	David, Andrew, Anna and Craig	Project Manager	Anna	discuss a new product, probably remote controller
2	2023-01-13 19:22:07	12	789314181	David, Andrew, Anna and Craig	Marketing	Craig	Nonsens
3		6	278952151	David, Andrew, Anna and Craig	Project Manager	Anna	No ifea
4		8	1770220038	David, Andrew, Anna and Craig	Designer	Anna	- Discuss the functional design: three new requirements have come in that need to be discussed: 1) teletext should be removed since it's outdated 2) control only TV, not DVD, VCR etc 3) The company wants the corporate colour and slogan to be implemented in the new design
5	2023-01-20 10:50:50	12	1110774290	David, Andrew, Anna and Craig	Designer	Anna	- discuss functional design, 3 new features: 1) remove teletext 2) RC should only control TV, not DVD, VCR, ... 3) implement corporate colour and slogan
6	2023-01-21 15:32:09	12	250634440	David, Andrew, Anna and Craig	Designer	Anna	discuss the um functional design
7	2023-01-27 11:30:45	12	1828459154	David, Andrew, Anna and Craig	Project Manager	Anna	ASDF
8		5	1519664016	David, Andrew, Anna and Craig	Designer	Anna	test
9	2023-01-27 12:12:42	12	1061964708	David, Andrew, Anna and Craig	Designer	Andrew	clickworker test
10		1	235787155	Peter, Bob, Steffanie and Curt			
11		1	434697458	Peter, Bob, Steffanie and Curt			
12		1	453655152	Mark, Robert, Sarah and Chris			
13		1	388223088	Peter, Bob, Steffanie and Curt			
14		1	1004251647	Peter, Bob, Steffanie and Curt			
15		1	409629127	Mark, Robert, Sarah and Chris			
16		1	252257218	Peter, Bob, Steffanie and Curt			
17		1	905256028	Mark, Robert, Sarah and Chris			
18	2023-01-27 14:10:07	12	1980404318	David, Andrew, Anna and Craig	Marketing	Anna	A discussion
19		1	632114422	Peter, Bob, Steffanie and Curt			
20		1	1105110529	Peter, Bob, Steffanie and Curt			
21			970060157				
22		0	1118334142				
23		1	49871129	Peter, Bob, Steffanie and Curt			
24		1	1015697169	Peter, Bob, Steffanie and Curt			
25	2023-01-27 14:37:16	12	917493802	David, Andrew, Anna and Craig	Project Manager	Anna	Marketing
26	2023-01-27 15:42:03	12	177898971	David, Andrew, Anna and Craig	Project Manager	Craig	Research about marketing
27		1	445225663	Peter, Bob, Steffanie and Curt			
28		1	1988862120	Mark, Robert, Sarah and Chris			
29	2023-01-27 15:23:50	12	930920510	David, Andrew, Anna and Craig	Project Manager	Craig	How to manage project
30	2023-01-27 15:53:07	12	1656051964	David, Andrew, Anna and Craig	Project Manager	Anna	An agenda is a list of meeting activities in the order in which they are to be taken up, beginning with the call to order and ending with adjournment. It usually includes one or more specific items of business to be acted upon. It may, but is not required to, include specific times for one or more activities
31		1	2080687797	Peter, Bob, Steffanie and Curt			
32		1	620921669	Peter, Bob, Steffanie and Curt			
33		1	303388085	Mark, Robert, Sarah and Chris			
34	2023-01-27 16:17:14	12	1495697691	David, Andrew, Anna and Craig	Marketing	Andrew	Selles and services
35		1	244619329	David, Andrew, Anna and Craig			
36		4	1863127593	David, Andrew, Anna and Craig	Project Manager	Anna	Function of the remote control.
37	2023-01-27 18:08:32	12	1703167202	David, Andrew, Anna and Craig	Project Manager	Anna	New project on remote control
38		1	95507140	Mark, Robert, Sarah and Chris			
39		1	2018156567	Peter, Bob, Steffanie and Curt			
40	2023-01-27 18:35:50	12	481949418	David, Andrew, Anna and Craig	Project Manager	Andrew	Yes
41		4	1300003285	David, Andrew, Anna and Craig	Project Manager	Andrew	How to complete the project on time. how well the project can be finished.
42		1	554819770	David, Andrew, Anna and Craig	Project Manager		
43		1	116156426	Mark, Robert, Sarah and Chris			
44		1	533580606	Mark, Robert, Sarah and Chris			
45		1	1470950638	Peter, Bob, Steffanie and Curt			
46		1	609130801	Peter, Bob, Steffanie and Curt			
47		1	50790146	Mark, Robert, Sarah and Chris			
48	2023-01-27 20:52:07	12	1456414692	David, Andrew, Anna and Craig	Marketing	Anna	how to market and advertising products
49		1	1572876417	Peter, Bob, Steffanie and Curt			
50	2023-01-27 23:21:17	12	256748015	David, Andrew, Anna and Craig	Project Manager	Anna	It's about Teletext, control of tv and design.
51	2023-02-04 16:26:26	12	822796269	David, Andrew, Anna and Craig	Designer	Anna	It is to discuss the functional design of the remote control
52	2023-02-05 11:46:59	12	2015956562	David, Andrew, Anna and Craig	Designer	Anna	discuss functional design of the new remote control that they are working on
53		1	245250110	Mark, Robert, Sarah and Chris			
54	2023-02-06 10:28:18	12	536225802	David, Andrew, Anna and Craig	Designer	Anna	clickworker test
55		1	1992580262	Mark, Robert, Sarah and Chris			
56		11	625310324	David, Andrew, Anna and Craig	Marketing	Anna	conducting an academic user test about meeting summarization
57	2023-02-06 17:15:17	12	531011429	David, Andrew, Anna and Craig	Marketing	Anna	conducting an academic user test about meeting summarization
58		1	1400590?2	Mark, Robert, Sarah and Chris			
59	2023-02-06 17:44:46	12	1015891190	David, Andrew, Anna and Craig	Designer	Anna	There was no clear agenda presented, the meeting was to discuss the functionality of the remote control
60	2023-02-06 17:31:55	12	1579433742	David, Andrew, Anna and Craig	Project Manager	Craig	To discuss the features of a remote control product.
61	2023-02-06 17:29:09	12	831162968	David, Andrew, Anna and Craig	Marketing	Anna	desire to make their remote control product fashionable and sleek
62	2023-02-06 17:56:28	12	529376609	David, Andrew, Anna and Craig	Designer	Anna	Discoussion of the functional design of the remote control
63	2023-02-06 17:43:52	12	1701131936	David, Andrew, Anna and Craig	Marketing	Anna	conducting an academic user test about meeting summarization
64		3	59754879	David, Andrew, Anna and Craig	Marketing	Andrew	
65		1	344273889	Peter, Bob, Steffanie and Curt			
66		1	1743236219	Peter, Bob, Steffanie and Curt			
67		1	433869809	Peter, Bob, Steffanie and Curt			
68	2023-02-06 19:09:58	12	359509472	David, Andrew, Anna and Craig	Marketing	Anna	To discuss a new remote
69	2023-02-06 19:12:57	12	1003765759	David, Andrew, Anna and Craig	Designer	Anna	To discuss the design of new remote control
70	2023-02-06 22:27:06	12	805456022	David, Andrew, Anna and Craig	Project Manager	Andrew	To distribute the work and get ideas.
71		1	1051281645	Mark, Robert, Sarah and Chris			
72	2023-02-07 8:35:05	12	748419111	David, Andrew, Anna and Craig	Designer	Anna	General discussion about the remote

Question 5	Question 6	Question 7
a very complicated one with lots of buttons and colours and a simple one with buttons and labels	...the advanced functions which I quite like having on the controls...	Between 15 and 35
		Between 15 and 35
Unclear		
a very complicated one that's got lots of buttons , lots of colours , very confusing you , don't know what you're doing versus a very simple one	- Users dislike the look and feel of current remote controls ==> ugly - The vast majority would spend more money for it to look fancy as well we'll , see later , the vast majority would spend more money for slightly more intuitive control , such as voice recognition - most people would uh adults at least would pay more for voice recognition	Between 15 and 35
very complex with many buttons and colours versus basic but clunky	- users dislike the look and feel of current remote controls . So they find them ugly . Most people find them ugly - The vast majority would spend more money for it to look fancy as well we'll , see later , the vast majority would spend more money for slightly more intuitive control , such as voice recognition - Um most people use only a very slim portion of all the controls Summary: So I guess what we're looking at here is people want this technology , they tend to use the most simple controls and overall they find remote controls to be something that doesn't really appeal to them	Between 15 and 35
there's , either um a very complicated one that's got lots of buttons , lots of colours , very confusing you , don't know what you're doing and there is a very simple one.	They want a fashionable remote, because users dislike the look and feel of current remote controls . So they find them ugly. It has to be simple	Between 15 and 35
ASDF	ASDF	40 and above
test	test	Between 15 and 35
Too sensitive	Less buttons	40 and above
		Between 15 and 35
Researching market and analyst	Customer need good quality remote control tools. The required for a simple home theater illustrated the problems caused by complexity and inconsistency in user interface.	Between 15 and 35
Have voice command feature and shortcut menu	More customized button	Between 15 and 35
Fewer people need a universal remote control these days, which explains why the category is dying and great options are hard to find. But if your home-entertainment system is more complex than the basic media player–TV–soundbar combo, and you're looking for one remote to control all your gear, the SofaBaton U1 Universal Remote Control is the best option we've found. It has some notable flaws, but this model can control a wider variety of home-entertainment devices, and it has a better physical design than its competitors.	Remote access is the ability for an authorized person to access a computer or network from a geographical distance through a network connection.	Between 15 and 35
Fewer people need a universal remote control these days, which explains why the category is dying and great options are hard to find.	We made use of a myriad of resources for our research. First we browsed the World Wide Web with several popular search engines. These resulting web sites supplied the initial background information. Specific company web sites and online product manuals, however, showed us the current products on the market. The U.S. government patent site supplied similar designs that	Between 15 and 35
High cost and low quality	Long lasting	40 and above
Talk and curiosity	Good quality product yes,users looking for in a remote control.	Between 15 and 35
mistakes made during last set-up no immediate access	allow operation of devices that are out of convinient reach for direct operation of controls	Between 15 and 35
There two extremes 1 very complicated one with lots of buttons and 2nd colours and a simple one with buttons and labels.	customer group and corresponding pricing	40 and above
a very complicated one with lots of buttons and colours and a simple one with buttons and labels	Users dislike the look and feel of current remote controls. Most only use a very slim portion of all the controls. People want something to look fancy and they want technology.	Between 15 and 35
very complex one with lots of buttons and a very basic clunky one	They are looking for fancy design, intuitive control	Between 15 and 35
test	test	Between 15 and 35
a very complicated one with lots of button and colours and a simple one with buttons and labels	customer needs the best remote control for their tv	Between 15 and 35
a very complicated one with lots of button and colours and a simple one with buttons and labels	desire to make their products fashionable and sleek	Between 15 and 35
a very complicated one with lots of buttons and colours, and a simple one with buttons and labels	something that that looks good and is easy to use , but has fairly powerful product features	Between 15 and 35
Complicated with lots of buttons.	Simple, intuitive controls.	Between 15 and 35
a very complicated one with lots of button and colors and a simple one with buttons and labels	customer needs the best remote control for their tv and they went more complicated product	Between 15 and 35
a very complicated one with lots of buttons and colours and a simple one with buttons and labels	Many people use only a small portion of the remote control, something that that looks good and is easy to use , but has fairly powerful product features	Between 15 and 35
a very complicated one with lots of button and colors and a simple one with buttons and labels	customer needs best remote controls for there Tv also they went more complicated products	Between 15 and 35
Complicated and simple	Many people on use a small portion of the remote. Needs to be easy to learn	Between 15 and 35
Price and bit clunky	Price range and new functionality	Between 15 and 35
Volum and start /stop	Volume controls.	40 and above
The lack of buttons and difficulty of use	Ease of use and less buttons	40 and above

Question 8	Question 9	Question 10
they are used to high tech devices not as people over 40 and people below 20 do not have money to buy this	16 Euros	power source, Has to change channels , has to change volume
Silly	16 Euros	Easy handling an prty data mining
They are used to technology and have some expendable income		
young professionals with affinity to technology and some money to spare	25 Euros	change channle, volume, power
those are young professionals because they have bit of expendable income to spend on this sort of thing	25 Euros	Has to change channels , has to change volume, a good power source, control the television
ASDF	25 Euros	ASDF
test	16 Euros	test
Most people	16 Euros	Less difficulty operating
15 to 35	25 Euros	Marketing of product
Identified a target market allows marketers to focus on those most likely to purchase the products.	25 Euros	*Distance facilities cler, ongoing communication between our designers and client, which builds trust over time.
The target group is more often interact with devices, because of their affinity to technology	25 Euros	To access devices
Few brands only appeal to one specific demographic of people; one age group or gender, most companies will need to appeal to a much wider variety.	25 Euros	Running a great design process is hard enough when you're in an office. Working remotely can feel like dialing up that difficulty from very hard to expert, especially at first. (Our fully-distributed team of 23 can attest to that.) Once you get the hang of it, though, we've found that remote work makes for a better design process. All you need are a few ground rules, a virtual whiteboard, and the following lessons that we learned the hard way. Here's hoping you won't have to.
Target market segmentation is the process of dividing your target market into smaller, more specific groups. It allows you to create a more relevant marketing	25 Euros	Once you get the hang of it, though, we've found that remote work makes for a better design process. All you need are a few ground rules, a virtual whiteboard, and the following lessons that we learned the hard way.
They use more remote because of health problems	16 Euros	To control from a distance
This target group was selected for quality work	16 Euros	structure and size
because they actively have more engagement and screen time out and use computer and tech	25 Euros	ensures clear ongoing communication between designers and clients and this builds trust over time
so they aiming at a fairly young market, young professionals	25 Euros	Volume and channel and, skip to specific channels with the numbers.
Those are the people who use the computer in their everyday work and are willing to try to use technology	25 Euros	Has to change channels and volume and needs a power source
young professionals that have expendable income to spend on this kind of thing and are willing to use technology	25 Euros	Has to change channels and volume
test	25 Euros	test
grouping of controls according to use cases	16 Euros	desire to make their products fashionable and sleek
grouping of controls according to use cases	16 Euros	change channel and volume
Sort of young professionals, kind of	25 Euros	Has to change channels , has to change volume
Less reliant on computers and have expendable income	25 Euros	Brightness/contrast, on/off, channel select, teletext.
grouping of controls according to use cases	25 Euros	a very complicated one with lots of button and colors and a simple one with buttons and labels
you want it's somebody who's not gonna just use the remote that comes with their TV	25 Euros	Has to change channels , has to change volume
the target group is used to technology	16 Euros	Has to change channels , has to change volume
Because they use a computer daily	25 Euros	change volume and channel
They are picked as they have significantly high income, money to spare	25 Euros	Channel and volume change
They pick the experienced one	25 Euros	start /stop
It's the older generation	16 Euros	Less buttons

Question 11	Review
low end, middle, high end	maybe a advanced ctrl+f search function in the transcript which also searches for synonyms of the input chosen by a certain bench mark.
1 2 2003	Nothing
One would be audio controls , one would be video controls , and the other one would be device controls	Please note: if you measure the time taken to answer the survey, I actually had to restart because I forgot to save - so I was quicker on some questions than I would have been. I liked using the analyser and having concrete questions. The summary sentences are pretty good! My suggestions: - would be cool if the audio was accessible, too - make the summary sentences and the topics searchable (for specific questions like this it would be super helpful) - the transcript context when you click on a summary sentence is very handy, but I would add two things: 1) ability to click to previous/next segment (often I was in roughly the right part of the transcript to answer the question, but it was not there yet, so I had to click out of the context and on the next summary sentence ... it was hard to remember which one it actually was :) 2) Show the corresponding summary sentence in the title next to "Transcript Context" - ok, not sure how this would work with my previous suggestion, where probably the summary sentences would change as you click through the transcript context... - As I said, the summary sentences are really good, but there are quite a lot - so maybe they could be grouped together / tagged with a topic, e.g. "functional design", "pricing", ...
the habitual ones that should be right within your natural grip and others that are uh also available and then others that are concealed	selecting on the timeline in the right upper corner opens the respective frames in the summary sentences. back & close buttons
ASDF	ASDF
test	test
Design, buttons and controls	Is a great app
Marketing	Experience
Platform, ground based, airborne	Improve sell and service. Insure good quality and price.
Channel, Apps Shortcut, Configruation	It's already excellent
There are three primary areas or classifications of security controls. These include management security, operational security, and physical security controls. What is Management	Improve services. Insure good quality and sell.And also good and reasonable price.
In electronics, a remote control (also known as a remote or clicker[1]) is an electronic device used to operate another device from a distance, usually wirelessly	It's really good
Short medium and long	File not opening
structure,size, technically	I feel very good and yes I would improve.
IR based systems, RD based systems, and BT based systems	quite great experience to improve on distance
controll devicde and audio video	Overall is brilliantly designed, just needs a search bar to find the keyword in it.
Audio controls, video controls and device controls	Fantastic
audio, video, device	Nothing
test	test
speaker network, technical, remote control	
audio, video	The experience was amazing
One would be audio controls , one would be video controls , and the other one would be device controls	There was inconsistency between using the keywords box and the summary sentences to find the information I was looking for. The keywords did not always provide the content i was seeking so I ended up using the summary sentences to be sure.
Not sure.	I liked the experience, did not see anything to improve on.
speaker network, technical, remote control	Yes i like the experience also the app is good
audio controls , one would be video controls , and the other one would be device controls	The experience was okay. I am missing a search function. Even if its only for the key words
audio, video and device controls	The experience was great and the app design is likeble
Audio, display, often used	Didn't fit the screen on my phone too well in portrait mode.
Audio, video, device	It is very slow
integrate, remotes and connectivity	I like the data they shared but it takes too much time to load. It's not stable enough.
Small, Large and Medium	Less buttons

Q1 Time	Q2 Time	Q3 Time	Q4 Time	Q 5 Time	Q6 Time	Q7 Time	Q8 Time	Q9 Time	Q10 Time	Q11 Time
20.39	49.56	150.72	85.57	109.31	145.99	135.25	61.62	183.73	204.49	73.51
26.02	6.95	7.54	17.63	32.07	5.44	8.88	17.03	7.7	40.3	31.55
6.85	32.95	3.19	20.5	8.43	2.55					
13.29	146.61	47.03	200.37	158.76	209.94	60.39	72.74			
3.65	3.59	3.57	89.78	83.23	134.97	3.52	79.12	16.36	79.32	40.1
88.64	143.85	71.92	159.63	164.43	270.41	48.31	158.3	24.2	128.18	197.95
3	2.15	2.56	2.5	2.27	2.28	3.38	2.53	3.45	2.69	3.23
15.12	7.95	2.82	4.66	3.09						
4.9	5.24	3.42	4.91	3.48	2.75	4.44	2.91	17.87	4.21	3.35
3.62										
4.83										
2.72										
6.4										
4.96										
10.52										
5.6										
5.83										
5.28	3.07	2.72	10.95	15.99	18.85	5.33	8.17	13.82	20.14	17.57
80.63										
28.94										
13.2										
42.74										
249.7										
4.87	9.24	7.68	10.66	26.79	8.5	31.74	15.47	16.95	39.6	30.57
9.76	31.24	23.85	30.58	180.79	286.97	101.7	232.52	176.47	231	258.18
27.45										
3.66										
3.76	9.61	10.8	56.14	112.98	69.97	20.72	44.83	47.1	87.96	41.81
8.47	33.64	13.98	205.32	121.12	88.67	18.76	110.23	73.08	86.65	143.45
4.43										
4.58										
4.68										
6.24	92.12	83.32	208.33	154.55	13.62	90	298.59	17.44	63.75	108.53
57.49	18.21									
27.45	15.16	7.06	50.56							
7.95	5.13	22.74	26.68	55.69	15.53	4.58	18.41	17.61	13.43	27.71
27.54										
5.45										
3.47	16.77	10.61	42.86	214.67	137.3	14.16	121.88	23.7	79.42	120.44
34.29	11.57	11.81	99.83							
14.58	229.59									
12.24										
12.74										
18.27										
27.2										
10.27										
5.67	14.52	9.82	61.8	168.56	110.72	10.57	36.83	62.68	170.87	81.56
207.63										
157.49	169.88	29.11	110.38	161.87	80.18	124.99	115.38	235.74	118.63	81.45
15.31	21.02	35.7	20.54	43.9	164.3	33.93	42.96	44.47	51.79	39.33
11.36	121.81	10.09	25.83	65.78	75.36	22.09	67.33	12	42.76	21.04
7.91										
3.05	6.97	17.56	9.25	3.17	3.04	5.3	3.04	22.79	3.55	3.36
13.93										
3.49	11.51	94.27	147.79	177.95	341.52	6.17	67.36	25.08	76.47	47.23
3.65	6.8	3.86	34.17	304.09	45.94	79.62	39.29	55.32	211.44	89.55
7.9										
23.45	85.22	60.06	250.74	177.33	66.53	86.96	38.4	204.27	69.25	132.84
8.34	71.8	40.87	35.87	65.97	58.19	7.61	36.27	5.98	41.5	66.22
3.27	4.38	4.15	35.05	16.42	40.5	29.31	12.11	24.86	39.28	53.18
57.77	186.38	111.55	117.93	27.14	181.19	49.62	46.34	14.5	71.73	114.21
15.27	3.63	3.48	8.73	27.59	70.02	12.74	38.16	13.31	9.27	10.54
10.62	96.8	28.02								
5.21										
5.54										
2.5										
9.01	33.06	31.54	11.19	57.59	83.13	6.11	123.18	17.31	148.35	160.23
3.34	3.59	3.99	15.99	57.1	18.9	4.33	18.77	14.21	16.02	14.35
96.8	9.77	11.48	153.09	63.61	75.65	49.08	27.06	13.16	16.83	87.15
12.59										
4.47	3.37	2.81	17.65	29.82	15.28	5.27	11.51	14.32	9.02	20.65

B.0.3 Transcript users

The following tables show the answers of the users utilizing the transcript for the user test.

Id	Submit date	Lastpage	Seed	Question 1	Question 2	Question 3	Question 4
1	1/27/2023	11	1162048406	David, Andrew, Anna and Craig	Marketing	Craig	Marketing
2	1/27/2023	11	80769933	Peter, Bob, Steffanie and Curt	Project Manager	Anna	Analysis
3	1/27/2023	11	714754470	Mark, Robert, Sarah and Chris	Project Manager	Andrew	Increasing their productivity
4		3	702307296	Mark, Robert, Sarah and Chris	Designer	Anna	
5	1/27/2023	11	112184480	David, Andrew, Anna and Craig	Project Manager	Anna	New project
6	1/27/2023	11	1262006983	David, Andrew, Anna and Craig	Designer	Anna	Functional design
7		1	1134902740	David, Andrew, Anna and Craig			
8	1/27/2023	11	993496000	David, Andrew, Anna and Craig	Marketing	Anna	Marketing
9		3	15144062	Mark, Robert, Sarah and Chris	Project Manager	Andrew	
10		1	861805842	David, Andrew, Anna and Craig			
11	1/28/2023	11	1609623678	Peter, Bob, Steffanie and Curt	Designer	Anna	How to create a perfect room for a surveyors
12	1/28/2023	11	2038628748	David, Andrew, Anna and Craig	Designer	Anna	The agenda is about a remote control whose functional design should be revised.
13	1/28/2023	11	1762669997	David, Andrew, Anna and Craig	Designer	Anna	Discuss the functionality of a new remote control.
14	1/28/2023	11	204268532	David, Andrew, Anna and Craig	Project Manager	Anna	functional design
15	1/28/2023	11	1586185364	David, Andrew, Anna and Craig	Marketing	Anna	It was about Teletext and Internet. Now a days no one is using teletext much and mostly they use it only for TV remote. So it's about the design of the product.
16	1/28/2023	11	1056429303	David, Andrew, Anna and Craig	Designer	Anna	To design remote control for the TV as per customer needs and also to focus on latest design
17	2/4/2023	11	1469101823	David, Andrew, Anna and Craig	Designer	Craig	Discuss the functional design of the remote control
18			668366663				
19	2/6/2023	11	1897482189	David, Andrew, Anna and Craig	Project Manager	Craig	Discussing a products design.
20		3	1081190726	David, Andrew, Anna and Craig	Marketing		
21	2/6/2023	11	592841923	David, Andrew, Anna and Craig	Designer	Anna	there was no agenda
22	2/6/2023	11	1868946726	David, Andrew, Anna and Craig	Designer	Craig	Recap of previous meeting Cost 3 new requirements - teletext is outdated due to the internet, product design (in terms of appearance, corporate colors etc), technical design finer details of the remote control (& lunch!)
23	2/6/2023	11	1801713876	David, Andrew, Anna and Craig	Project Manager	Anna	To discuss the functional design.
24	2/6/2023	11	1972870322	David, Andrew, Anna and Craig	Designer	Anna	Designing a remote thing
25	2/6/2023	11	119641174	David, Andrew, Anna and Craig	Marketing	Anna	Discuss functional design of a remote control

Question 5	Question 6	Question 7
Marketing Product	Marketing	Between 15 and 35
Meeting, analysis		Between 15 and 35
Buttons and range	smart remotes/universal remote. One can work for many devices and be durable.	Between 15 and 35
Positive and negative	To control from a distance	40 and above
To control the teletext. The remote control was used to control the television not , the VCR DVD or anything else.	Few buttons on the remote control. Basic functions like switch on/off, volume up/down	40 and above
No idea	Resources at a cheaper raye	Between 15 and 35
In order not to enter the wrong key	In able to use for their appliances	Between 15 and 35
High-tech on the one hand, simple on the other.	Users want a nice looking control, wich they can use intuitive and in that case, it should have all functions they need.	40 and above
One extreme is a complicated design with lots of buttons and colours, and poor labelling. The other extreme was a simple design with less buttons, but still the same size as the more complex one.	A intuitive control with voice recognition, high tech but still easy to use, good aesthetic and ergonomics.	Between 15 and 35
control the television not , the VCR , DVD or anything else	fashion in electronics	Between 15 and 35
Volume and channel controls are up and down or to be in numbers.	To change channel, volume and may be to adjust screen resolutions (brightness etc)	Between 15 and 35
Channels and volume controls and its design	To change channel, volume and screen resolution (brightness, colour etc)	Between 15 and 35
Two extremes: - A very complicated one that has a lot of buttons and colors and is confusing - A very simple one with only basic functions which is clunky	Many users would spend more money for a fancy looking remote with intuitive controls such as voice recognition	Between 15 and 35
Clunky buttons and tricky features.	Mainly on and off buttons and brightness or contrast control.	Between 15 and 35
a very complicated one that's got lots of buttons , lots of colours , very confusing - It's a very simple one . It's got only the basic functions	users dislike the look and feel of current remote controls the vast majority would spend more money for slightly more intuitive control , such as voice recognition	Between 15 and 35
Transmission technology of the remote - infra red or something else batteries - standard ones or a rechargeable unit	Cost - if it gets lost/broke not to expensive to replace basic controls - change channel/volume/power etc voice control power consumption - batteries or a rechargeable unit ease of use - seperateing out the common functions along with the more specialized functions	Between 15 and 35
Either a very complicated one with lots of buttons and colours that is very confusing or a very simple one with only the basic functions.	They want it to look more fancy 'fashionable, sleek and trendy) and more intuitive (like voice recognition).	Between 15 and 35
Chunky and price	Voice recognition and price	Between 15 and 35
Controlling DVD players and VCRs, and communicating with Teletext	Simplicity, and the brand on the remote.	Between 15 and 35

Question 8	Question 9	Question 10	Question 11
15 to 35	25 Euros	Marketing	Marketing, product lunching, marketing of product
Accuracy	10 Euros	Give access to a desktop protocol	Location, domain
most people wantting to watch tv or do stuff using remote control fall under that category	10 Euros	on off volume up and down and menu	IR based wifi based and bluetooth
They use more remote because of health problems	10 Euros	To control from a distance	Small distance, medium distance and long distance
most people would uh adults at least would pay more for voice recognition	25 Euros	speech recognition	15-35, 25, 40 and above
They occupy a large population	25 Euros	No idea	No Idea
Group B	10 Euros	The functional keys	Start, control and off
Because they want to add include speech recognition and think, that this target group would response high to that.	25 Euros	- Volume - Channel switch - On and off - Channel switch with numbers	video, audio and device controls
This age group has expendable money to spend on new technology, and would typically be using a computer every day at work.	25 Euros	Voice recognition, on off up and down (channels and volume), mute.	Audio, video, and device.
aiming at a fairly young marke	25 Euros	basic functions are the logic , the transmitter , and the receiver	One would be audio controls , one would be video controls , and the other one would be device controls
To discuss about the design of the remote which is used currently only for TV controls. And the ways to design it as simple as possible with the needed features	25 Euros	Basic functions are to change channels, volume and may be to adjust screen resolution.	Volume, channel and screen resolution and it should have the logo colour and verses
This age group ppl may use remote a lot compared to other age groups in the discussion	25 Euros	Channel, volume and screen resolution	Logo colour, slogan and design
Those people have a bit of expendable income and are willing to try new technology and use computer in their everyday work	25 Euros	On, off, up down, switch channels and volume	audio controls, video controls and device controls
Not dependent on computers.	15 Euros	On/off, volume, brightness.	Not sure
they have bit of expendable income to spend on this sort of thing	25 Euros	the logic , the transmitter , and the receiver	One would be audio controls , one would be video controls , and the other one would be device controls
disposable income willingness to try new technology products people older than 40 may not be willing to try it as not as used to the technology - target age range is more "gadgtety"	25 Euros	On , off , up , and down , channels up and down volume and skip to a channel .	Audio, Video & Device
By including voice recognition, they need to target a little older group than 25. They have a bit of expendable income to spend on this sort of things. It's people that use the computer in their everyday work. It's people who are gadgety	25 Euros	ON/OFF. Switch the channel up and down. Volume up and down. numbers for the channels	The habitual ones that are within your natural grip, others also available and then others concealed
Targeted at young professionals	25 Euros	Voice recognition and remote control	Voice recognition, volume and channel
Because of voice recognition technology	25 Euros	Volume control, power on off, channels up and down, skip to a channel.	Audio controls, device controls, video controls.

Q1 Time	Q2 Time	Q3 Time	Q4 Time	Q 5 Time	Q6 Time	Q7 Time	Q8 Time	Q9 Time	Q10 Time	Q11 Time
9.49	3.86	4.26	9.99	19.38	12.79	4	12.33	8.71	9.36	49.14
3.92	3.4	6.28	23.61	22.4	15.74	6.89	19.85	4.68	38.51	118.73
256.82	8.54	11.03	37.76	233.87	152.86	8.8	44.74	43.95	63.35	104.59
82.36	69.17	5.8								
8.7	7.39	4.85	10.48	17.69	26.27	8.18	30.48	8.27	17.51	32.42
5.3	110.52	6.28	125.43	190.81	718.3	473.88	64.33	39.27	102.47	416.58
12.33	115.83									
33.75	11.65	14.48	14.67	15.79	170.38	7.88	25.88	9.48	8.38	12.41
5.27	7.44	4.79								
42.82										
103.54	48.42	220.95	813.66	55.48	46.41	30.47	132.98	17.26	23.2	39.2
7.83	181.96	4.35	486.52	385.44	146.05	39.06	241.62	113.81	178.24	139.13
28.12	169.46	6.04	246.51	160.75	241.36	22.64	57.32	35.4	210.83	29.85
18.14	2020.26	38.36	128.45	59.4	48.72	96.7	36.09	18.63	27.98	52.08
11.52	26.87	23.17	94.37	58.15	58.14	19.59	122.34	39.63	47.03	83.73
8.52	11.15	4.54	50.51	38.43	34.91	20.15	74.63	21.43	22.22	32.2
11.46	76.37	10.54	73.66	105.83	112.63	103.44	38.07	17.89	96.55	47.14
17.52	81.9	16.99	97.09	38.34	62	32.14	144.03	151.27	92.18	222.63
144.23	127.37	105.32	9.86							
26.2	117.74	104.29	242.28	152.05	139.62	111.45	38.38	4.09	54.74	27.71
18.15	88.76	38.42	212.57	92.6	121.03	28.8	97.68	12.29	34.89	127
138.55	327.78	107.16	304.37	129.27	223.3	76.43	253.27	59.18	89.8	143.82
26	104.56	20.8	13.47	168.84	65.75	58.33	24.8	5.68	23.94	233
9.46	151.52	12.96	37.95	51.76	56.94	130.91	39.36	54.67	216.49	35.18