



Sebastian Schäffer, Bsc

Summarizing Long Scientific Documents: Leveraging Llama2-7B-Chat with Explainable AI

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Co-Supervisors

Dipl.-Ing Sarah Frank, Dipl.-Ing. Dr.techn Andreas Wagner

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Frank Kappe

In cooperation with
CERN
Geneva, Switzerland

Graz, February 2024



Sebastian Schäffer, BSc

Summarizing Long Scientific Documents: Leveraging Llama2-7B-Chat with Explainable AI

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Co-Supervisors

Dipl.-Ing Sarah Frank, Dipl.-Ing. Dr.techn Andreas Wagner

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Frank Kappe

In cooperation with
CERN
Geneva, Switzerland

Graz, February 2024



Sebastian Schäffer, Bsc

Zusammenfassen langer wissenschaftlicher Artikel: Einsatz von Llama2-7B-Chat mit erklärbarer KI

Masterarbeit

zur Erlangung des akademischen Grades eines

Diplom-Ingenieur

Masterstudium: Software Engineering and Management

eingereicht an der

Technische Universität Graz

Betreuer

Assoc.Prof. Dipl.-Ing. Dr.techn. Christian Gütl

Mitbetreuer

Dipl.-Ing Sarah Frank, Dipl.-Ing. Dr.techn Andreas Wagner

Institute of Interactive Systems and Data Science
Vorstand: Univ.-Prof. Dipl.-Ing. Dr.techn. Frank Kappe

In Zusammenarbeit mit
CERN
Genf, Schweiz

Graz, Februar 2024

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift

Abstract

In an era where literature is growing enormously, skimming and understanding long documents is more important than ever. This is especially true for the scientific literature, which is expanding tremendously daily. Traditional methods of manual text summarization are not only time-consuming but also vulnerable to individual bias.

This thesis focuses on developing and evaluating an advanced AI-based system for summarizing long scientific documents, emphasizing accuracy, coherence, and transparency. The system utilizes state-of-the-art models such as Meta’s large language model Llama2-7B-Chat with an NVIDIA A100 Tensor Core GPU, providing a robust foundation for efficient summaries. A significant feature of this system is integrating an explanation mechanism based on the SBERT-based all-MiniLM-L6-v2 model, designed to increase user trust by linking sentences from the summary back to the corresponding detailed content in the original document. The system’s graphical user interface provides an intuitive and easily accessible way of user interaction.

The effectiveness of the system was evaluated in two studies. These evaluations, which involved authors of scientific papers and a diverse group of 34 participants, assessed the system’s performance in terms of accuracy, coherence, and the impact of explainability on user trust. The results show high satisfaction with the system’s ability to accurately reproduce scientific papers’ content and highlight the most important research contributions. The system’s effectiveness, which allows users to trace the content of the summaries back to the original document, save time, and create an appropriate length of summaries, was also rated well. However, there is room for improvement in coherence and the consistent enhancement of user trust through transparency. This research underlines the importance of explainability in AI systems. By exploring the intersection of Natural Language Processing and Explainable AI, this work contributes to the field of automatic text summarization by providing a solution to the challenge of understanding large amounts of scientific literature.

Kurzfassung

In einer Zeit, in der die Menge an Literatur enorm wächst, ist das Überfliegen und Verstehen langer Dokumente wichtiger denn je. Dies gilt insbesondere für die wissenschaftliche Literatur, die täglich immens wächst. Herkömmliche Methoden der manuellen Textzusammenfassung sind nicht nur zeitaufwändig, sondern auch anfällig für individuelle Voreingenommenheit.

Diese Arbeit fokussiert sich auf die Entwicklung und Evaluierung eines fortschrittlichen KI-basierten Systems zur Zusammenfassung langer wissenschaftlicher Dokumente, wobei der Schwerpunkt auf Genauigkeit, Kohärenz und Transparenz liegt. Das System nutzt modernste Modelle wie das große Sprachmodell Llama2-7B-Chat von Meta mit einer NVIDIA A100 Tensor Core GPU und bietet so eine solide Grundlage für effiziente Zusammenfassungen. Ein wesentliches Merkmal dieses Systems ist die Integration eines Erklärungsmechanismus, der auf dem SBERT-basierten all-MiniLM-L6-v2-Modell aufbaut und darauf ausgelegt ist, das Vertrauen der Benutzer zu erhöhen, indem Sätze aus der Zusammenfassung mit den entsprechenden detaillierten Inhalten im Originaldokument verknüpft werden. Die grafische Benutzeroberfläche des Systems bietet eine intuitive und leicht zugängliche Möglichkeit der Benutzerinteraktion.

Die Wirksamkeit des Systems wurde in zwei Studien evaluiert. Diese Bewertungen, an denen Autoren wissenschaftlicher Arbeiten und eine vielfältige Gruppe von 34 Teilnehmern beteiligt waren, bewerteten die Leistung des Systems im Hinblick auf Genauigkeit, Kohärenz und den Einfluss der Erklärbarkeit auf das Benutzervertrauen. Die Ergebnisse zeigen ein hohes Maß an Zufriedenheit mit der Fähigkeit des Systems, den Inhalt wissenschaftlicher Arbeiten präzise wiederzugeben und die wichtigsten Forschungsbeiträge hervorzuheben. Gut bewertet wurde auch die Effektivität des Systems, das es den Nutzern ermöglicht, den Inhalt der Zusammenfassungen bis zum Originaldokument zurückzuverfolgen, Zeit zu sparen und Zusammenfassungen in angemessener Länge zu erstellen. Allerdings besteht Verbesserungspotential bei der Kohärenz und der consequenten Stärkung des Nutzervertrauens durch Transparenz. Diese Forschung

unterstreicht die Bedeutung der Erklärbarkeit in KI-Systemen. Durch die Erforschung der Schnittstelle zwischen Natürliche Sprachverarbeitung und erklärbarer KI trägt diese Arbeit zum Bereich der automatischen Textzusammenfassung bei, indem sie eine Lösung für die Herausforderung bietet, große Mengen wissenschaftlicher Literatur zu verstehen.

Acknowledgments

I would like to thank my family and friends for their support throughout my academic journey.

I am profoundly thankful to Christian Gütl for providing me with this opportunity, sharing his extensive expertise, and offering fast support throughout every stage of this thesis.

I also want to thank Sarah Frank for her constant feedback and support at CERN.

I want to thank Andreas Wagner for the opportunity to work at CERN and for general support.

Finally, I would like to thank Alexander Steinmaurer, Alexander Nussbaumer, and Igor Jakovljevic for their support and feedback.

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Contribution and Research Question	1
1.3. Thesis Structure	2
2. Background and Related Work	4
2.1. Fundamentals of Natural Language Processing	4
2.1.1. Historical Overview of NLP	5
2.1.2. Preprocessing in NLP	7
2.1.3. Core Applications in NLP	9
2.2. Automatic Text Summarization	11
2.2.1. Extractive Summarization	12
2.2.2. Abstractive Summarization	18
2.2.3. Datasets	23
2.2.4. Evaluation Metrics	27
2.3. Explainable Artificial Intelligence	31
2.3.1. Historical Overview of Explainable AI	32
2.3.2. Explanation Types	33
2.3.3. Explainability Techniques	34
2.3.4. Visualization Techniques	35
2.3.5. Evaluation Techniques	37
2.4. Related Work on Explainability for Text Summarization	40
2.5. Summary	43
3. Requirements and Concept	44
3.1. Functional Requirements	44
3.2. Non-Functional Requirements	46
3.3. Conceptual Architecture	47
3.4. Design Decisions	49
3.5. Summary	50
4. Development	51
4.1. Architecture	51
4.2. Development Details	54
4.2.1. Input Interface and Preprocessing	54

4.2.2. Summarization Module	56
4.2.3. Explainable AI Module	59
4.2.4. Graphical User Interface	63
4.3. Summary	65
5. Evaluation	66
5.1. Summarization Quality Study	66
5.1.1. Study Design	66
5.1.2. Setting and Instruments	67
5.1.3. Procedure	68
5.1.4. Results and Discussion	69
5.2. Explainability Study	74
5.2.1. Study Design	74
5.2.2. Setting and Instruments	74
5.2.3. Procedure	76
5.2.4. Results and Discussion	77
5.3. Limitations	83
5.4. Summary	84
6. Lessons Learned	85
6.1. Literature	85
6.2. Development	86
6.3. Evaluation	86
7. Conclusion and Future Work	88
7.1. Conclusion	88
7.2. Future Work	89
Bibliography	91
A. Screenshots of the system's output	103
A.1. Generated summaries and explainability	103

List of Figures

2.1.	Architecture of the extractive text summarization system incorporating statistical metrics (Ma et al., 2022)	15
2.2.	Scaled Dot-Product Attention in the Transformer's Architecture. Adapted from Vaswani et al. (2017).	22
2.3.	Two out of six ideal summaries consisting of 4 SCUs (Nenkova et al., 2007)	28
2.4.	An example of a decision tree for AI project strategy. Adapted from Quinlan (1990).	32
2.5.	Visualization of an attention heatmap showing the alignment between English source words and their French translations during neural machine translation. Adapted from Bahdanau et al. (2014).	36
3.1.	The Conceptual Architecture of the system	48
4.1.	The flowchart outlines the simplified architecture of the summarization system and highlights the core components and their connections to each other	53
4.2.	The input interface for uploading PDF documents to the summarization system, with drag-and-drop function	55
4.3.	Display of the interface after successful upload of the PDF document	55
4.4.	Summary of the BERT paper generated by the system	59
4.5.	Displaying the most similar sentence from the original BERT paper based on the user-selected summary sentence	62
4.6.	Overview of the design concept for the system's GUI	63
4.7.	Graphical User Interface of the system	64
5.1.	Age range of authors (Q1.1)	69
5.2.	Gender of authors (Q1.2)	70
5.3.	Highest degree or level of education (Q1.3)	70
5.4.	Profession of authors (Q1.4)	70
5.5.	Experience with Artificial Intelligence (AI) tools (Q1.5)	71
5.6.	Accuracy with which the summary reflects the content (Q1.6)	72

List of Figures

5.7. To what extent the summary highlights the key contributions (Q1.7)	72
5.8. Coherence of the AI-generated summary (Q1.8)	72
5.9. Overall satisfaction with the summary (Q1.9)	73
5.10. Appropriate length of the summary (Q1.10)	73
5.11. Age range of participants (Q2.1)	77
5.12. Gender of participants (Q2.2)	78
5.13. Highest degree or level of education (Q2.3)	78
5.14. Profession of participants (Q2.4)	78
5.15. Experience with Artificial Intelligence (AI) tools (Q2.5)	79
5.16. Clarity of the AI-generated summaries (Q2.6)	80
5.17. To what extent the user trusts the AI-generated summaries to be accurate and reliable (Q2.7)	80
5.18. Effectiveness of the system in allowing users to trace the summary content back to the original document (Q2.8)	80
5.19. Coherence of the AI-generated summary (Q2.9)	81
5.20. The extent to which the transparency of the system contributes to general trust in the system (Q2.10)	81
5.21. The extent to which the AI system's explainability function supports work interactions with summaries (Q2.11)	81
A.1. Summary 1	103
A.2. Selected Summary 1 Sentence	104
A.3. Most similar sentence in the original document 1	104
A.4. Summary 2	104
A.5. Selected Summary 2 Sentence	104
A.6. Most similar sentence A in the original document 2	105
A.7. Another selected Summary 2 Sentence	105
A.8. Most similar sentence B in the original document 2	105
A.9. Summary 3	105
A.10. Selected Summary 3 Sentence	106
A.11. Most similar sentence A in the original document 3	106
A.12. Another selected Summary 3 Sentence	106
A.13. Most similar sentence B in the original document 3	106

List of Tables

2.1.	Overview of types of summaries. Adapted from Nenkova and McKeown (2011); Jones (1998); Hovy and Lin (1998); Awasthi et al. (2021).	13
2.2.	Overview of the ten statistical features. Adapted from Fattah and Ren (2009).	14
2.3.	Overview of the rhetorical relations. Adapted from Mann and Thompson (1988).	16
2.4.	Overview of commonly used datasets for summarization systems.	26
2.5.	Overview of ROUGE Metrics. Adapted from C.-Y. Lin (2004).	30
2.6.	Overview of Explanation Types. Adapted from Arya et al. (2019).	34
2.7.	Key properties of AI explanations. Adapted from Carvalho et al. (2019).	38
5.1.	Overview of the authors' responses to the quantitative evaluation criteria.	73
5.2.	Overview of the participants' responses to the quantitative evaluation criteria.	82

1. Introduction

This chapter focuses on the motivation for this work. In addition, the contribution and structure of the thesis are defined.

1.1. Motivation

In an era where scientific literature is growing incredibly, the ability to skim lengthy documents is more important than ever. This challenge is further intensified by the need for clarity and depth in understanding these texts. Traditional manual summaries are time-consuming and subject to personal bias, highlighting the need for an automated system to produce accurate, coherent, and valuable summaries.

This work fulfills this requirement by using artificial intelligence to develop and evaluate a text summarization system that distills extensive scientific texts into concise summaries. With such an advanced AI summarization system, it is crucial that not only the developers understand the system to increase performance but also that the user can build more trust in the summaries created by the AI in the long term. For this reason, users can link each summarized sentence to its detailed source in the original document, increasing transparency and trust while demonstrating the application of Explainable AI. This system aims to assist professionals in quickly capturing the core ideas of long scientific papers. It uses state-of-the-art models to achieve this objective.

1.2. Contribution and Research Question

This work makes several contributions to the fields of NLP and Explainable AI. The main contribution is developing an explanation mechanism for an AI-based text summarization system. The system strengthens user confidence

by utilizing the SBERT-based all-MiniLM-L6-v2 model embeddings. It links sentences from the summary to the most similar semantic content in the original document. Furthermore, it uses state-of-the-art technologies, such as Meta’s large language model Llama2-7B-Chat with an NVIDIA A100 Tensor Core GPU, and optimizes the model for summarizing long scientific documents. In addition, the system’s architecture, designed for efficiency and scalability, integrates a user-friendly interface created with Gradio that enables broad accessibility.

Additionally, two comprehensive evaluations were conducted as part of this work to assess the quality of the summaries and the effectiveness of the system’s explanation feature, providing valuable feedback for further refinement. The primary research questions addressed in this thesis are:

- **RQ1:** How do authors rate the quality of the summaries produced by the AI system in terms of their accuracy, coherence, and overall usefulness for understanding the original scientific literature?
- **RQ2:** How effectively does the AI summary system provide transparency in its summary process, and how does this transparency impact user trust?

By addressing these questions, the work attempts to bridge the gap between advanced AI summarization capabilities and the need for transparency in AI systems.

1.3. Thesis Structure

Chapter 2 discusses the theoretical foundations of NLP, automatic text summarization, and the role of explainability in artificial intelligence. It also refers to relevant work on the explainability of text summaries.

Chapter 3 describes the functional and non-functional requirements for the system. Furthermore, it presents the conceptual architecture and design decisions behind its development.

Chapter 4 describes the technical architecture of the system, including the choice of technologies used, the development processes, and the implementation of the modules, such as the Summarization and Explainable AI modules.

Chapter 5 presents the methodology and results of the Summarization Quality Study and the Explainability Study, which provide insights into the system's performance and user feedback.

Chapter 6 provides an overview of the research, development as well as evaluation process and summarizes the most important findings, the challenges encountered, and how they were overcome.

Chapter 7 concludes the thesis with a summary of the results, contributions to the field, and suggestions for future research to improve the capabilities of the system and the user experience.

2. Background and Related Work

This chapter provides a foundational literature review that is relevant to this thesis and provides a comprehensive background for the following discussions. It starts by outlining critical developments in natural language processing (NLP), highlighting essential pre-processing techniques and primary applications such as part-of-speech tagging, named entity recognition, machine translation, text classification, and automatic text summarization that have shaped the field. The focus then shifts to automatic text summarization in Chapter 2.2, distinguishing between extractive and abstractive approaches and emphasizing the importance of datasets and evaluation metrics, which are essential for these methods. In addition, the chapter discusses the need for transparency in AI through explainable artificial intelligence (XAI). It highlights the historical development, the various explanation and visualization techniques, as well as the methods for evaluating the explainability of AI systems. To conclude, the intersection between explainability and text summarization is examined, and an insight into current research trends in this area is provided.

2.1. Fundamentals of Natural Language Processing

Natural language processing (NLP), a subfield of artificial intelligence, aims to bridge the gap between human communication patterns and computational methods, attempting to model and replicate the nuances of human linguistic behavior. As stated by Liddy (2001), *“Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications”*(p.2126).

2.1.1. Historical Overview of NLP

NLP started gaining importance as a field of study during the mid-20th century. It was in the 1950s when there emerged a notion of machines that could comprehend and generate human speech. One of the most significant achievements of this era was the presentation of Russian-English machine translation by IBM in collaboration with Georgetown University in New York City in January 1954. It immediately caught the public's attention and instantaneously sparked much discussion. The system had a vocabulary of 250 words and was able to apply six grammar rules to translate various sentences (Hutchins, 2004). This early attempt, notwithstanding the limited computing power of its time, indicated the enormous potential of machine translation and laid the foundation for future NLP research.

In 1957, Chomsky (1957) introduced a more structured theoretical approach in his book *"Syntactic Structures"*. A finite set of grammar rules can produce an infinite number of sentences, which underlines the generative ability of human language. The introduction of generative transformational grammar marked a paradigm shift in linguistics, which is now no longer primarily descriptive but very theoretically based (Chomsky, 1957). Following this theoretical breakthrough, a practical milestone was achieved through the computer program *"ELIZA"*, developed by Weizenbaum (1966), which simulates a Rogerian psychotherapist. The program uses decomposition rules triggered by keywords to parse input sentences and generates responses through corresponding reassembly rules.

The 1980s and 1990s marked a time of major change for NLP. The field gradually moved away from purely deterministic, rule-based methods and shifted towards statistical models. The development of a discrete dictation recognition program by Jelinek (1985) exemplified the potential of statistical methods. The statistical approach was a trigram language model constructed from a 25-million-word text database. The trigram model predicts the next word after a pair of words based on the sequences seen during the training (Jelinek, 1985). In the 1990s, this shift became even more noticeable as researchers were no longer constrained by the limitations of rule-based systems and began to explore the vast possibilities of statistical approaches. The peak of this trend was the influential book *"Foundations of Statistical Natural Language Processing"* by Manning and Schutze (1999) in 1999. This book has been essential for researchers and students entering the domain of NLP since its comprehensive coverage of statistical methods has made it a standard work in the field.

The dawn of the 21st century marked another transformative era for NLP since deep learning techniques heavily influenced it. One of the milestones of this period was the development of word embeddings, mainly through the Continuous Bag-of-Words (CBOW) model and the Continuous Skip-gram model, which Mikolov et al. (2013) presented. Instead of sparse, high-dimensional vectors, words were now represented in continuous vector spaces that captured rich semantic relationships. CBOW tries to predict the current word from its surrounding context words. For instance, for the sentence "Cats love to chase mice", with the given words "Cats", "love", "to" and "mice", it will try to predict the word "chase". The Skip-Gram model does the opposite by attempting, given the target word "chase," to predict the surrounding context, which includes words such as "cats," "love," "to," and "mice."

Starting in 2010, there was renewed interest in Recurrent Neural Networks (RNNs), which initially became popular in the 1990s through the Elman Network, introduced by Elman (1990). RNNs are designed to process sequential data by maintaining a state from previous inputs. For training, the RNNs use backpropagation, in which the gradients of the loss function are propagated backward through the neural network. During this process, the vanishing gradient or the exploding gradient may occur, and as a consequence, the RNNs cannot learn long-range dependencies in the data (Salehinejad et al., 2017). Long Short-Term Memory Networks (LSTMs), first proposed by Hochreiter and Schmidhuber (1997), became crucial to address this challenge. LSTMs have implemented gating mechanisms to capture long-term dependencies in data (Hochreiter & Schmidhuber, 1997).

Since 2014, Convolutional Neural Networks (CNNs), initially developed for image classification, have been applied in NLP. Kim (2014) explores the application of CNNs for sentence-level classification tasks by training a CNN with a convolutional layer over word vectors derived from an unsupervised neural language model. In the same year, the Sequence-to-Sequence (Seq2Seq) model was introduced by Sutskever et al. (2014). Using two LSTMs, one acting as an encoder to compress the inputs and the other as a decoder to produce the outputs, the Seq2Seq model architecture has led to breakthrough innovations in machine translation (Sutskever et al., 2014).

In 2017, the NLP community experienced a significant paradigm shift by introducing the Transformer architecture developed by Awasthi et al. (2021). The Transformer model, which uses an innovative self-attention mechanism, diverged from traditional architectures such as RNNs and CNNs and has rapidly gained acceptance for tasks such as machine translation.

The self-attention mechanism allows the model to weigh the relevance of various parts of the input data, thus making it also particularly efficient for automatic text summarization (Awasthi et al., 2021). The following year, the Bidirectional Encoder Representations from Transformers (BERT) model was introduced by Devlin et al. (2018). The novelty of BERT lies in its bidirectional training, which incorporates context from both sides of a word into its input. This bi-directionality, combined with the pre-training and fine-tuning using an additional output layer, allowed BERT to set new standards for various NLP tasks (Devlin et al., 2018). That same year, OpenAI introduced Generative Pre-trained Transformer 1 (GPT-1), developed by Radford et al. (2018), which uses the Transformer architecture. The model is trained on a large corpus of unlabeled text and then fine-tuned on a smaller, labeled dataset that depends on the task (Radford et al., 2018). The most significant 2019 variant of GPT-2, presented by Radford et al. (2019), had 1.5 billion parameters, about thirteen times larger than its predecessor. The model's capacity plays a critical role in its ability to solve new tasks without seeing examples during training, as increasing the capacity also improves performance (Radford et al., 2019). GPT-3, presented by Brown et al. (2020) in 2020, with its massive 175 billion parameters, showed superior performance compared to GPT-2 by excelling in few-shot settings and accomplishing various tasks without domain-specific training. OpenAI published the latest model, GPT-4, in March 2023. GPT-4 shows comparable performance to humans on various professional exams, a breakthrough in AI. It offers new possibilities and is vulnerable to multiple attacks, so it is vital not to neglect the security aspect (OpenAI, 2023).

2.1.2. Preprocessing in NLP

Natural language processing has made rapid progress in recent years, and careful preprocessing of textual data is a critical factor in the success of various NLP tasks. The preprocessing process involves a series of systematic operations that automatically process raw text, converting it into a more suitable format for machine learning algorithms. One main reason is the ability to reduce text data significantly. For instance, stop words are often redundant, and their removal can account for 20-30% of a document's total word count. Furthermore, stemming techniques can lead to a remarkable 40-50% reduction in the size required for data indexing (Kannan et al., 2014). The most common pre-processing steps are described in detail in the following paragraphs.

Tokenization involves dissecting text into smaller units, called tokens, like

words. It serves as the initial step in breaking down the text into its essential, meaningful components, allowing algorithms to analyze the composition of sentences more efficiently (Kannan et al., 2014). Although simple UNIX commands can provide preliminary word statistics, tokenization often requires more nuanced methods to address the complexity of the language structure. Implementing effective tokenization often relies on deterministic algorithms, usually based on regular expressions. The Python-based Natural Language Toolkit (NLTK) provides such functionality for tokenization. Byte pair encoding (BPE) is another method for tokenizing texts, which is very effective for handling unknown words in NLP. Rather than just recognizing words or characters, BPE creates subwords by analyzing and merging frequent pairs of adjacent symbols in a training corpus (Jurafsky & Martin, 2023). The difficulty of tokenization varies depending on the language. English and French, which use spaces to separate words, are more straightforward to tokenize than languages such as Chinese and Thai, which do not have clear word borders. The complexity of tokenization also depends on the writing systems and the underlying structure of words in the languages. For instance, while words in Chinese are not broken down into smaller units, in Japanese, this is the practice (Kannan et al., 2014).

Sentence segmentation is crucial in word processing, using punctuation marks such as full stops, exclamation points, or question marks to separate sentences. Periods are ambiguous because they either denote a sentence boundary or are part of an abbreviation such as "Inc". For this purpose, sentence and word tokenization can be approached together. Usually, sentence tokenization first uses rules or machine learning to determine whether a period is part of a word or signifies the end of a sentence (Jurafsky & Martin, 2023). Kiss and Strunk, 2006 introduced an unsupervised multilingual sentence boundary detection system that utilizes collocation evidence to detect abbreviations, ordinal numbers, and initials, thereby reducing dependence on orthography.

Stop words are high-frequency words such as "are" and "and" occur frequently in documents without adding significant meaning to the content. In text mining, they can hinder understanding the context of a document. Hence, their removal can improve system performance. However, building a consistent stop word list is challenging because of the differences between different text sources (Kannan et al., 2014). Removing stop words is not as helpful as once thought for many tasks. For instance, it made searching for phrases more difficult, as "to be or not to be" became just "not" (Jurafsky & Martin, 2023).

Stemming involves identifying the stem or root of a word. For instance, the

words "player", "players", "played", and "play" can all be stemmed to the root of "play". The main goal of this technique is to remove various suffixes, ensure consistent stem matches, reduce the word counts, and save both memory and processing time. Several algorithms facilitate this stemming process (Ramasubramanian & Ramya, 2013). The Lovins Stemmer, which Lovins (1968) proposed, was the first known stemmer. In this method, the longest suffix is removed from the words. After removing the suffix, the terms are adjusted using a table to form valid words. However, some suffixes are missing from the table, and occasionally, unreliable stems are produced (Lovins, 1968). The Porter stemming algorithm, introduced by Porter (1980), is considered one of the most popular stemming techniques. The algorithm includes five different phases. When applied to a text, words are converted to their base form using a set of rules. Different rules are checked at each stage, and the corresponding suffix is removed if a rule meets the criteria. This process proceeds through all five stages until the final word stem is obtained (Porter, 1980).

Lemmatization is the process of either eliminating or changing the suffix of a word to convert it to its base form, which is called a lemma. Unlike stemming, lemming is always meaningful. For example, the word "caring" is lemmatized to "care," which is a meaningful term (Tabassum & Patil, 2020). Lemmatization uses morphological parsing, which examines how words are formed from basic units called morphemes. Words can have multiple morphemes. For example, "cats" has two: "cat" and "-s" (Jurafsky & Martin, 2023). It uses vocabulary and morphology analysis to return words to their dictionary form, considering context, such as using verbs or nouns. Therefore, it can match synonyms, e.g., "hot" with "warm" or "car" with "automobile" (Balakrishnan & Lloyd-Yemoh, 2014).

2.1.3. Core Applications in NLP

The aim of this work is to focus on selected key concepts of NLP. Core methods of NLP are part-of-speech tagging, named entity recognition, text classification, machine translation, and automatic text summarization. The following sections discuss these critical areas in more detail, highlighting their importance and applications in the rapidly evolving landscape of Natural Language Processing.

In part-of-speech tagging (POS tagging), each word in a text is tagged with the corresponding word class. Given a set of words as input, the objective is to generate a corresponding set of part-of-speech tags as an output (Jurafsky

& Martin, 2023). Over the years, multiple techniques have been developed to address this challenge. Rule-based taggers operate using hand-written rules tailored to specific linguistic patterns of a language. The Brill Tagger, introduced by Brill (1992), starts with an initial set of manually created linguistic rules and then refines and optimizes these rules based on an annotated corpus. Instead of relying entirely on predefined rules, the tagger is designed to identify and correct its weaknesses (Brill, 1992). Probabilistic taggers calculate the probability that a tag is correct for a given word based on historical data. In this category, a hidden Markov model (HMM) that estimates tags based on tag sequence probabilities is widely favored (Brants, 2000). With the rise of Deep Learning, neural networks have also been used for POS tagging. Models such as recurrent neural networks have successfully captured complicated patterns without relying on manually created rules (Perez-Ortiz & Forcada, 2001).

Named Entity Recognition (NER) is the process of identifying sections of text that represent specific names and categorizing them based on the type of entity they represent. The four predominant entity categories are person, location, organization, and geopolitical entity. However, a named entity often encompasses more, such as dates, times, or numerical values, such as prices (Jurafsky & Martin, 2023). The first systems were based on manually created rules, which were domain-specific. They were used for financial messages, for example, which were adequate but not scalable (Farmakiotou et al., 2000). The advent of machine learning marked the transition to statistical models in NER. Algorithms such as Hidden Markov Model (HMM) were widely used for Named Entity Recognition (Morwal et al., 2012). In recent years, deep learning methods such as recurrent neural networks, convolutional neural networks, recursive neural networks, long short-term memory networks, and transformer-based models have achieved astonishing results in NER (Li et al., 2020).

Text classification (TC) divides text documents into predefined categories. There are various classifiers, including decision tree, rule-based, probabilistic, and neural network classifiers. Each classifier has its own strengths and weaknesses that make it applicable to specific tasks (Vijayan et al., 2017). One of the most common tasks within TK is sentiment analysis, which assesses the sentiment in text data such as tweets or product reviews. It determines whether the sentiment is positive, negative, or, in more sophisticated usage, possibly a spectrum of emotions. Topic analysis identifies the main topics of a text and news categorization and organizes news content according to user preferences. Other tasks involve question answering, which evaluates the accuracy of an answer based on the text provided, and natural language inference predicts whether the meaning of one text can be inferred from

another (Minaee et al., 2021).

Machine Translation (MT) involves utilizing computers to convert text from one language to another. MT first began with rule-based methods, which relied heavily on the expertise of linguists, who accurately created dictionaries and established grammatical rules to support the translation process between languages (Shiwen & Xiaojing, 2014). Statistical Machine Translation (SMT) emerged with the availability of bilingual corpora. SMT considers natural language translation as a machine learning challenge. By studying countless human translations, SMT systems learn to translate autonomously (Lopez, 2008). The introduction of Neural Machine Translation (NMT) represented a significant milestone. Unlike traditional statistical machine translation, neural machine translation focuses on creating a neural network optimized to improve translation quality (Bahdanau et al., 2014). Further tremendous progress was made with the Transformer architecture, which set new standards in machine translation (Vaswani et al., 2017).

Automatic Text Summarization aims to generate a concise summary of a text without losing its core concepts (Ježek & Steinberger, 2008). This complex task is addressed in the following chapters, which provide comprehensive insights into the methods used to accomplish it effectively. The next section focuses, in particular, on extractive and abstract text summarization, the two most important summarization methods.

2.2. Automatic Text Summarization

The amount of data on the internet has increased exponentially in the last ten years. This rise requires strategies to transform this enormous volume of raw data into meaningful, straightforward information for humans to understand. Much research in recent years has focused on automatic text summarization, which is one of the well-known research methods to process these vast amounts of data. Automatic text summarization distills a document's essential concepts by preserving important information and creating a condensed text version (Awasthi et al., 2021).

There are different types of summaries: The two main approaches of automatic text summarization are Extractive Summarization and Abstractive Summarization. An extractive summarizer uses the sources' most important sentences as input to create a summary. In contrast, an abstractive summarizer processes the source's main idea and develops a summary using

paraphrasing. Furthermore, it is necessary to differentiate between multi-document summarization, where the input consists of multiple documents on a common topic, and single-document summarization, where only one source is utilized, such as a scientific paper or a news article (Nenkova & McKeown, 2011).

It is also possible to categorize a summary as indicative or informative. An indicative summary means the reader's attention is drawn without revealing the details. An informative summary, on the other hand, summarizes the most critical content (Jones, 1998). For instance, (Malyusz et al., 2021) state, "Algorithm 5/c is the fastest for Monte Carlo simulation among the studied two-phase algorithms. On average, it performs sixteen times better than Algorithm#2, almost fifty times better than Algorithm#3, and around 250 times better than Algorithm#5." The indicative summary could be: "Among the evaluated two-phase algorithms for Monte Carlo simulation, Algorithm 5/c proves to be the most efficient." In contrast, the informative summary could read as follows: "In Monte Carlo simulation tests among the two-phase algorithms, Algorithm 5/c performed better than the others and was sixteen times faster than Algorithm#2, fifty times faster than Algorithm#3, and about 250 times faster than Algorithm#5."

In addition, it is essential to differentiate between query-oriented and generic summaries. A query-oriented summary focuses on certain parts of the input text to respond precisely to the user's direct input, producing a summary tailored to the user. By contrast, a generic summary uses no prior knowledge to make a summary (Hovy & Lin, 1998). Apart from that, supervised and unsupervised methods are vital for the selection of sentences in the final summary. This distinction is crucial since supervised methods require data to train the model, whereas unsupervised methods do not (Awasthi et al., 2021). Table 2.1 shows an overview of the previously discussed types of summaries.

In the following sections, this thesis will discuss extractive summarization as well as abstractive summarization in detail. Following that, datasets and different evaluation metrics for automatic text summarizations will be presented.

2.2.1. Extractive Summarization

Extractive summarization focuses on selecting and extracting entire sentences or phrases from the source document to create a summary. Rather

2. Background and Related Work

Table 2.1.: Overview of types of summaries. Adapted from Nenkova and McKeown (2011); Jones (1998); Hovy and Lin (1998); Awasthi et al. (2021).

Summarization technique	Abstractive	Generates new sentences
	Extractive	Uses existing sentences
Number of documents	Single	One input document
	Multi	Multiple input documents
Summary purpose	Indicative	Presents main topic
	Informative	Focuses on critical points
User-specificity	Query-oriented	Tailored to a specific request
	Generic	General overview
Learning method	Supervised	Learns from labeled data
	Unsupervised	Detects patterns independently

than creating new sentences as in abstractive summarization, extractive summarization identifies and extracts portions of the original content that are considered most important and relevant (Gambhir & Gupta, 2017).

Most extractive text summarization methods can be divided into specific classes. This work presents approaches based on statistics, discourse, topics, graphs, and machine learning.

Some of the first methods for automatic summarization were statistical methods because they are mathematically simple and do not require extensive computational resources. For instance, if a particular term occurs frequently in a document, it is likely significant, making term frequency a straightforward and intuitive measure. The statistical method aims to select sentences based on their concrete properties and not on their meaning or the relationships between words. According to Fattah and Ren (2009), there are ten different statistical features, as depicted in Table 2.2.

2. Background and Related Work

Table 2.2.: Overview of the ten statistical features. Adapted from Fattah and Ren (2009).

Position	The order of the sentences is crucial because the first sentences of a paragraph are typically the most significant
Centrality	Sentence centrality determines the use of the same vocabulary in sentences throughout the document
Resemblance to the title	The sentence's resemblance to the title is measured by the shared word usage between the sentence and the title
Proper nouns	The sentence with more proper nouns will most likely be included in the summary
Positive key-words	Positive keywords in a sentence are those that often appear in the summary
Negative keywords	Negative keywords in the sentence are not likely to be used in the summary
Numerical data	The sentence containing numerical data will most likely be included in the summary
Relative length	The sentences that do not fulfill the minimum sentence length requirement are excluded from the summary
Bushy path of the node	The sentence with many links connecting a sentence to other sentences is likely to be included in the summary
Aggregate similarity	The aggregate similarity is measured by summing the similarities of links to other sentences

The architecture of the extractive text summarization system using statistical metrics is depicted in 2.1. This approach involves several stages, starting with pre-processing to prepare the texts for the summary process. Activities such as the removal of stop words, the tokenization of sentences, lemmatization, and sentence segmentation are part of this process. The system then proceeds to create a text representation. In this phase, the previously processed text is converted into a format, for instance, vectors, which enables simple analysis. Once the text is presented in a suitable format, the sentence-scoring phase begins. In this step, each sentence is evaluated based on various statistical features, which may include elements such as centrality or position in the document. After the evaluation, the system continues extracting sentences with high scores. The sentences with the highest relevance scores are selected and included in the summary, as it is assumed that they contain the most important topics of the source text. Finally, the process ends with post-processing, where the extracted key sentences are further refined to enhance the coherence of the final summary (Ma et al., 2022).

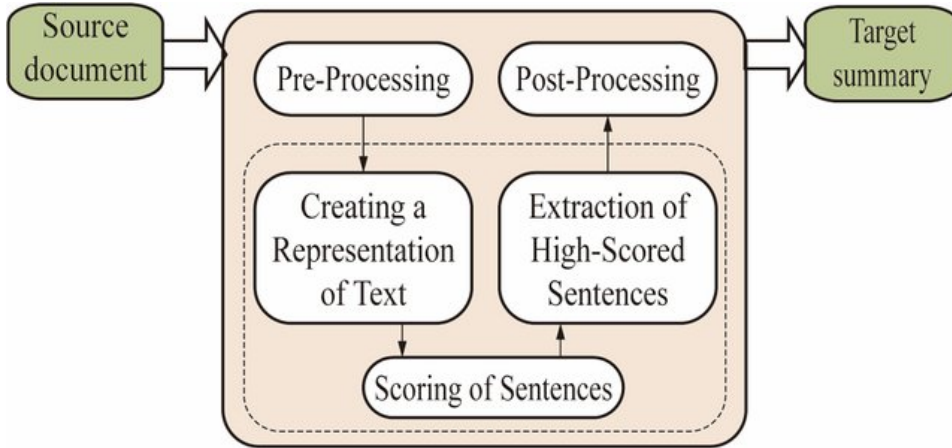


Figure 2.1.: Architecture of the extractive text summarization system incorporating statistical metrics (Ma et al., 2022)

In the late 1980s, approaches such as the Rhetorical Structure Theory (RST), developed by Mann and Thompson (1988), emerged that focused attention on the discourse structures in the text. This theory aims to ensure that the summary maintains the coherence and flow of the original content. The initial step is dividing the text into units, often corresponding to sentences or clauses. The authors use the term "span" to indicate that these units of text can vary in size and are not limited to sentences or clauses. After that, each pair of related spans is assigned a rhetorical relation, which describes their link. In each rhetorical relationship, one of the units is usually more central and is therefore referred to as the 'nucleus', while the other is less important and is referred to as the 'satellite'. The nucleus is what the discourse is about, and the satellite provides additional information about the nucleus. Therefore, the nucleus does not depend on the satellite, but the relationship between the two is necessary to obtain a coherent text. An overview of all the rhetorical relations mentioned in the paper can be seen in Table 2.3 (Mann & Thompson, 1988).

2. Background and Related Work

Table 2.3.: Overview of the rhetorical relations. Adapted from Mann and Thompson (1988).

Evidence	The nucleus makes a statement, and the satellite serves as empirical support
Justify	The nucleus presents a statement, and the satellite provides a justification for it
Antithesis	The nucleus makes a statement, and the satellite directly contradicts it
Concession	The nucleus presents a main argument, while the satellite takes up the counterargument
Circumstance	The nucleus describes the event, and the satellite outlines the context in which it takes place
Solutionhood	The nucleus presents a solution, and the satellite states the problem that is being solved
Elaboration	The nucleus makes a statement, and the satellite provides additional details
Background	The nucleus presents a primary message, and the satellite provides the background information that helps the reader grasp it
Enablement	The nucleus describes the event, and the satellite sets the conditions that make it possible
Motivation	The nucleus describes the event, and the satellite provides incentives for it
Volitional Cause	The nucleus describes the event that results from a volitional action, and the satellite describes the cause
Non-Volitional Cause	The nucleus describes the event that results from an unintended action, and the satellite describes the cause
Volitional Result	The nucleus describes a volitional action, and the satellite describes the result
Non-Volitional Result	The nucleus describes an action that is not volitional, and the satellite describes the result
Purpose	The nucleus describes an event, and the satellite outlines the objective behind it
Condition	The nucleus describes an event, and the satellite specifies the conditions under which it would occur
Otherwise	The nucleus sets the conditions, and the satellite describes what happens if they are unmet
Interpretation	The nucleus presents the event, and the satellite provides an interpretation
Evaluation	The nucleus describes the actions, and the satellite offers an assessment of it
Restatement	The nucleus makes a statement, and the satellite paraphrases it
Summary	The nucleus presents the content, and the satellite provides a brief summary
Sequence	This multinuclear relation involves a succession relationship between the situations represented in the different nuclei
Contrast	This multinuclear relation involves nuclei presenting opposing situations
Joint	This multinuclear relation involves nuclei that appear in the same text but are unrelated

As computational capabilities increased, topic modeling techniques such as Latent Semantic Analysis (LSA), introduced by Deerwester et al. (1990), and Latent Dirichlet Allocation (LDA), proposed by Blei et al. (2003), evolved. Topic modeling extracts the most relevant subjects from a text data collection. LSA is used to discover the latent semantic structures between terms and documents in a corpus. Initially, a term-document matrix is created, where each row represents a particular word, and each column represents a particular document. Each entry in this matrix indicates the frequency of a specific word in a specific document. Next, LSA performs Singular Value Decomposition (SVD) to decompose this large matrix into three smaller matrices. This process effectively reduces the data's complexity by preserving only the most essential topics. These topics serve as a reduced-dimensional space where words and documents can be represented. Similar words and documents are grouped in this compressed space, facilitating the identification of relationships between them (Deerwester et al., 1990). However, it is crucial to recognize the limitations of LSA. Since it is not a probabilistic model, it does not provide a probability distribution over topics, as with other topic modeling techniques such as Latent Dirichlet Allocation.

The primary purpose of LDA is to find the hidden thematic structure in a large text corpus. The model bases itself on the idea that each document is a mixture of several topics and every word originates from one of these topics. Each document can be represented as a distribution of topics and each topic as a distribution of words. This distribution not only helps to summarize the document but also simplifies the comparison and classification of documents based on their thematic content. As a result, the model can derive the likelihood of individual terms in relation to each topic and provide a quantifiable metric for the importance of features within each latent topic (Blei et al., 2003).

After introducing algorithms such as TextRank, developed by Mihalcea and Tarau (2004), and LexRank, proposed by Erkan and Radev (2004), graph-based methods gained widespread popularity. TextRank is an unsupervised and graph-based ranking algorithm that is suitable for extractive text summarization. The algorithm is based on the PageRank algorithm, which was presented by Page et al. (1998). In TextRank, a document is conceived as a graph in which each sentence is a node. The edges among these nodes are created using similarity measures, often calculated using techniques such as cosine similarity, longest common subsequence, or string kernel. Once this graph is built, PageRank scores each node on its connections and then ranks the nodes according to their importance within the graph. Finally, the nodes with the highest scores can be used to summarise the original text. This approach is particularly effective because it does not require training

data and can, therefore, be easily adapted to different text types (Mihalcea & Tarau, 2004). LexRank is another unsupervised graph-based technique developed for extractive text summarisation. The algorithm also represents a document as a graph with sentences as nodes. However, there are some significant differences to TextRank. It uses idf-modified cosine similarity to determine edges, which means that words that are common in the corpus but not necessarily informative in the specific document being summarised are given less weight. Furthermore, LexRank uses the eigenvector centrality to rank the sentences. The underlying principle of eigenvector centrality is that sentences that are like other important sentences capture the central topics of the document and should be considered more critical. LexRank is well suited for processing large amounts of data because of its computational efficiency (Erkan & Radev, 2004).

2.2.2. Abstractive Summarization

Contrary to extractive summarization, where key segments are taken directly from the source, abstract summarization introduces new words. Effective abstract summarization captures the critical points of the input while maintaining linguistic flow (Zhang et al., 2020).

Abstract text summarization usually involves three sequential processes: information extraction, content selection, and surface realization. Information extraction is about retrieving essential details from a text. One approach focuses on retrieving phrasal data, such as noun and verb phrases, along with their context (H. Lin & Ng, 2019). Another is query-based extraction to highlight important content and filter out less relevant information (Mehdad et al., 2014). Content selection is a crucial step in abstract text summarization, where relevant phrases are selected from those extracted during the information extraction phase to be included in the final summary. The phrases are often selected considering length constraints. In surface realization, the phrases selected in the content selection phase are merged into a summary using grammatical rules (H. Lin & Ng, 2019). Abstract text summarization methods can be broadly divided into three areas, which are structure-based methods, semantic methods, and deep learning methods (Rane & Govilkar, 2019).

A structured approach that uses a rule-based scheme is illustrated in the work of Genest and Lapalme (2012). The study aims to summarize content within domains such as "Accidents and Natural Disasters," "Attacks, Health and Safety," "Endangered Resources," and "Investigations/Trials". In the

information extraction phase, the authors employ a rule-based custom IE module tailored to the needs of abstract summarization. Based on abstraction schemas, this IE system extracts relevant information using hand-crafted rules. These schemas deal with different topics, such as "attacks," and aim to answer enough aspects of a given category to provide the necessary data needed to create the summary. The content selection process aims to select the most relevant information from the extracted data using a basic heuristic that selects the most frequently mentioned candidate for each aspect (Genest & Lapalme, 2012). Ultimately, SimpleNLG Realizer implements generation patterns for each schema to ensure the summary is concise and fluid (Gatt & Reiter, 2009). The GISTEXTER system developed by Harabagiu and Lacatusu (2002) is an example of a structured approach that uses topical relationships from WordNet and employs a set of ad hoc templates to create summaries. The creation of these templates is tailored to the topic that is to be summarized. The templates are filled using information extracted from the text, and each slot corresponds to a semantic role relevant to the topic at hand. By providing a structured way to capture and organize the most pertinent information from the text, these templates enable the creation of coherent and semantically relevant summaries (Harabagiu & Lacatusu, 2002).

One of the first working models for a semantics-based approach was the work of Genest and Lapalme (2011). The authors introduce the notion of Information Items (INITs) to form an abstract representation, seen as the tiniest coherent informational unit within a text. The creation of INITs is limited to subject-verb-object (SVO) triples that are both dated and localized. These INITs are then used to generate new sentences that form the summary (Genest & Lapalme, 2011). In the work of Moawad and Aref (2012), a semantic-based method uses a semantic graph. The authors propose a structured methodology that works in three main phases. First, the Rich Semantic Graph (RSG) is created from the source document. Second, the RSG is reduced to a more abstract form. Finally, an abstract summary is made from the reduced graph. In RSG, verbs and nouns from the source document are represented as nodes in the graph, with edges symbolizing their semantic and topological connections. The phase known as The Rich Semantic Graph Creation Phase aims at streamlining the originally formed semantic graph of the document into a more reduced version. A set of heuristic rules is used to reduce the size of the graph by either replacing, removing, or merging nodes using WordNet relationships. The Summarized Text Generation phase of generating the summarized text focuses on deriving the abstract summary from the reduced graph (Moawad & Aref, 2012).

Numerous deep-learning models are utilized for abstractive text summarization. Sequence-to-sequence models, considered one of the most fundamental deep learning approaches to abstract text summarization, use recurrent neural networks (RNNs) to process sequential data. RNNs process sequential inputs by maintaining a state that captures information from previous inputs (Song et al., 2019). The models are trained using backpropagation, a technique in which the gradients of the loss function are fed back through the network to adjust the weights. As a result, they may encounter problems such as vanishing or exploding gradients. The vanishing gradient problem occurs when the gradients shrink exponentially during backpropagation, while the exploding gradient problem occurs when the gradients grow exponentially. Consequently, this hinders the model’s ability to learn and retain long-term dependencies in the data.(A. H. Ribeiro et al., 2020).

To address these challenges, two significant gated variants of RNNs have been developed: Long Short-Term Memory (LSTM), developed by Hochreiter and Schmidhuber (1997), and Gated Recurrent Units (GRU), introduced by Cho et al. (2014). The unique solution these RNNs offer involves the introduction of gates into the network architecture. These gates determine the flow and processing of information within the network. Each gate is equipped with weights and biases that are fine-tuned during training. This allows these gates to selectively control the information passed on to the next stage or retained in the current state to maintain the gradient strength over long sequences. LSTM models contain complex gates, including input, memory, forget, and output. The input gate initiates its process with a vector that is initialized randomly. For the following steps, it employs the output from the memory cell of the preceding step as its input. The forget gate, a single-layer neural network equipped with a sigmoid activation function, decides whether the information of the previous state should be kept or eliminated. The memory gate consists of two neural networks and plays a role in determining the effect of stored information on newly acquired information. The first of these networks is like the forget gate but differs in the bias setting, while the second network uses a tanh activation function to generate new information. The output gate is crucial for forwarding new information to the subsequent LSTM unit. It works with a sigmoidal activation function that considers the input vector, the newly formed information, the previous hidden state, and the bias. The final output is obtained by multiplying the sigmoidal output of this gate by the tanh value of the newly generated information (Suleiman & Awajan, 2020). GRUs are designed with two specific gates, the reset and update gates, and do not include a separate memory unit (Cho et al., 2014). This design makes GRUs more effective at processing information from previous hidden states and allows them to compute faster than LSTMs. Nevertheless, LSTMs have the advantage

of more detailed control due to their particular memory unit (Song et al., 2019).

The RNN encoder-decoder model, developed by Cho et al. (2014), extends the capabilities of RNNs for situations where input and output sequences are of different lengths. This model consists of two components: An encoder, which converts a series of symbols into a fixed-length vector, and a decoder, which converts this vector back into a series of symbols. Its unique feature lies in its ability to manage sequences of different lengths and map them to another set, which can also be of different lengths. In addition to its use in abstractive text summarization, this encoder-decoder framework is also valuable for machine translation, where translated sentences often vary in length (Cho et al., 2014).

A significant progress in improving the learning process of sequence-to-sequence models was the introduction of the attention mechanism, originally applied in machine translation (Bahdanau et al., 2014). Later, Rush et al. (2015) presented a data-driven method for abstract text summarization that uses a model based on local attention. This involves creating each word in the summary based on the given sentence. The attention mechanism makes it possible for the model to concentrate on different parts of the input sentence at each step of creating the output. The weights in the attention mechanism indicate essentially the relevance of each input word. This model enables end-to-end training and is scalable for large training datasets (Rush et al., 2015). The model that Chopra et al. (2016) developed further develops the previously presented model. Instead of using a feed-forward neural language model to create summaries, this new model uses an RNN. Furthermore, the encoder in this updated model is more advanced as it considers the input words' position information (Chopra et al., 2016).

The introduction of the transformer model, proposed by Vaswani et al. (2017), has considerably influenced recent advances in abstract text summarization. Transformers focus on the concept of parallel processing of complex information and an advanced form of attention mechanism known as multi-head attention, which replaces the traditional recurrent layers in encoder-decoder structures. For instance, consider a sentence where different attention mechanisms could be applied to examine different aspects simultaneously. The structure of the transformer consists of multiple layers of encoders and decoders. These layers consist of units for attention and feedforward processes. There are six layers of encoder units stacked on each other and a corresponding stack of six decoder units. Each encoder layer has a multi-head attention component and a feedforward element. The decoder layers add to this by including a multi-head attention component and an

additional masked multi-head attention function in addition to the feed-forward element. The decoder focuses on all parts of the input sequence by referring to the encoder's output, similar to standard sequence-to-sequence models. In the encoder, self-attention layers allow each segment to consider the previous output, creating a connected processing flow. Similarly, the decoder's self-attention layers ensure that each step considers all previous steps while consciously avoiding any impact from future steps to maintain correct sequence generation (Vaswani et al., 2017). Figure 2.2 presents the scaled dot-product attention mechanism of the Transformer model's architecture.

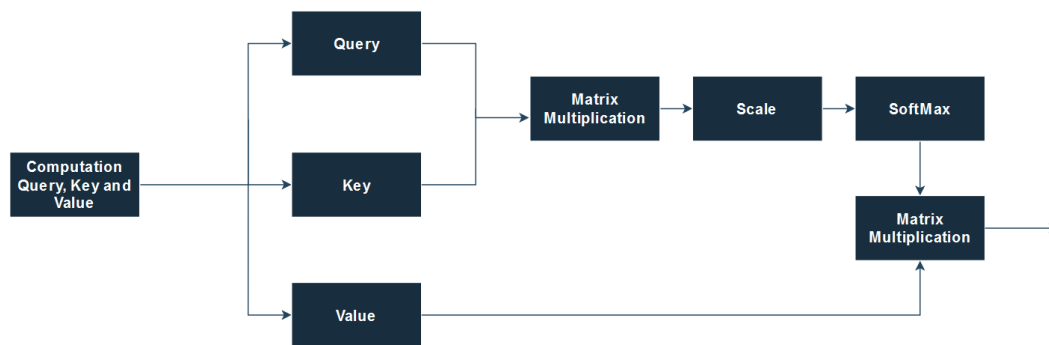


Figure 2.2.: Scaled Dot-Product Attention in the Transformer's Architecture. Adapted from Vaswani et al. (2017).

First, three vectors are derived from the input embeddings: Query, Key, and Value. A dot product between the key and query vectors determines the raw attention scores, a similarity measure. These scores are then adjusted by dividing them by the square root of the dimensionality of the key vectors. This prevents the next stage's softmax function from having a small gradient, which can occur with large dot products. The softmax function processes the scores to convert them into a probability distribution. These probabilities are the attention weights used to weight the value vectors accordingly. The resulting output attention vector is calculated simultaneously for each head. These vectors are then concatenated, linearly transformed, and fed into the feedforward network (Vaswani et al., 2017).

Google's BERT and OpenAI's GPT are both based on the Transformer architecture and pre-train on large-scale text data to tackle various abstract text summarization tasks. BERT is characterized by its ability to understand context in both directions simultaneously, as opposed to GPT, which processes text in a single direction (Devlin et al., 2018). BERT is a multi-layer encoder, while GPT is a multi-layer transformer-decoder (Radford et al., 2018). Meta's

BART, introduced by (Lewis et al., 2019), is also built on a Transformer architecture and extends the capabilities of BERT with its bidirectional encoder and GPT-like unidirectional decoder, as well as other pre-training methods. During pre-training, BART is exposed to intentionally corrupted text and learns to reconstruct the original, unmodified text. This model has proven successful in automatic text summarization tasks (Lewis et al., 2019). Many other transformer-based models have achieved remarkable success in various abstract text summarization tasks, such as PEGASUS (Zhang et al., 2020), T5 (Raffel et al., 2020), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023) and Llama 2 (Touvron et al., 2023).

2.2.3. Datasets

Datasets used for training and evaluation contribute significantly to the efficiency of an automatic text summarization algorithm because of their quality and nature. Different datasets are used for various purposes, such as abstract or extractive summarization. Often, they are also used in a domain-specific context and, therefore, play a fundamental role in shaping the capabilities of automatic text summarization systems. In the following sections, this work will address some of the most prominent datasets that have gained acceptance in the field.

The evaluation of summary methods conducted during the Document Understanding Conference (DUC) organized by NIST from 2000 to 2007 has provided a collection of annotated data sets. The DUC datasets train and evaluate text summarization systems, focusing on assessing general and specific summaries of English newspapers and newswire articles (Over et al., 2007). For instance, the 2003 corpus contains 624 document-summary pairs, each consisting of an original document and a summary of that document, and the 2004 corpus contains 500 pairs. The DUC corpora are commonly used to evaluate many existing abstract summarization programs. However, these relatively small corpora do not provide the extensive data generally required for training neural models (Nallapati et al., 2016). The Text Analysis Conference (TAC) grew out of the Document Understanding Conference (DUC) to promote community-wide evaluations of Natural Language Processing (NLP) technologies. TAC was launched in 2008 to build on the foundations of DUC in automatic text summarization, broadening its scope and fostering research within the NLP community. This has been achieved by providing the necessary assessment infrastructure, promoting research based on sizeable standard test collections, and stimulating the exchange of research ideas between industry, academia, and government

(NIST, 2023).

The Gigaword dataset, assembled by the Linguistic Data Consortium (LDC), includes a variety of English-language newswire documents (Graff et al., 2003). Gigaword comprises approximately 9.5 million news articles from various national and international sources over the past two decades. The dataset was first used by Rush et al. (2015)) for abstractive summaries. Each article’s headline and first sentence were paired for training to form input-summary pairs. Due to numerous misleading title-article pairs in Gigaword, heuristic filters were applied to refine the training set, resulting in approximately four million title-article pairs after filtering (Rush et al., 2015).

Already mentioned datasets like Gigaword and DUC only provide summaries of single sentences for automatic text summarization. In contrast, the CNN/Daily Mail dataset, introduced by Nallapati et al. (2016), is a robust collection that includes multi-sentence summaries. This corpus was created by modifying an existing corpus developed by Hermann et al. (2015) and initially used for passage-based question-answer tasks. In the original paper, the authors used summary bullet points from CNN and Daily Mail news stories to frame questions, using the stories as passages to answer those questions. With a minimal script adjustment, all bullet points for each story were rearranged in their original order, creating a multi-sentence summary in which each bullet point is considered a separate sentence. The resulting corpus includes 286,817 training pairs, 13,368 validation pairs, and 11,487 test pairs. Two versions of the dataset were published. The first one contains the actual names of the entities, and in the second one, the entities are replaced by document-specific integer IDs, starting with 0 (Nallapati et al., 2016).

This work aims to use scientific papers for the dataset. They are very convenient to use because they are significantly longer than news articles, and each one contains an abstract summary by its author. Combining the datasets arXiv and PubMed is popular for training and evaluating abstractive summarization models (Cohan et al., 2018). The arXiv dataset comprises a vast collection of scientific papers across various fields like physics, mathematics, and computer science, and the PubMed dataset consists of biomedical research papers.

The AMI Meeting Corpus, developed by Carletta et al. (2005), is a comprehensive multimodal dataset containing 100 hours of recorded meetings. It was developed as part of a project to improve meeting browsing technology and was, therefore, intended for publication. The corpus includes both

naturally occurring and staged meetings. In the latter, participants took on different roles within a design team and followed a project to develop a new remote control from start to finish. The dataset includes data on devices such as the near-field and far-field microphones and individual and room cameras (Carletta et al., 2005).

Remarkable corpora have emerged over the last few years, such as the BookCorpus dataset used to train Google’s language representation model BERT and GPT-N models (Bandy & Vincent, 2021). BookCorpus consists of 11,038 books taken from the Internet. These books are freely available and were written by people who had not published them then. The criterion was set to at least 20,000 words per book to exclude potentially shorter narratives, including books from 16 different genres (Y. Zhu et al., 2015). Other datasets worth mentioning are the Chinese corpus LCSTS (Hu et al., 2015), the IELTS summary corpus (Fang & Teufel, 2016), and BookSum, which is a compilation of datasets designed for the summarization of long-form narratives (Kryściński et al., 2021). Table 2.4 shows popular datasets that are used to train and evaluate summarization systems.

Table 2.4.: Overview of commonly used datasets for summarization systems.

Dataset	Number of Documents	Topic Scope	URL	Source
arXiv	215,913	Scientific papers	https://huggingface.co/datasets/scientific_papers	Cohan et al. (2018)
CNN/Daily Mail	311,971	News	https://huggingface.co/datasets/ccdv/cnn_dailymail	Hermann et al. (2015)
DUC (2001-2007)	Varies	News	https://www-nlpir.nist.gov/projects/duc/data.html	Over et al. (2007)
Gigaword	9,876,086	News	https://catalog.ldc.upenn.edu/LDC2011T07	Graff et al. (2003))
TAC	Varies	News	https://tac.nist.gov/data/index.html	NIST (2023)
PubMed	133,215	Scientific papers	https://huggingface.co/datasets/scientific_papers	Cohan et al. (2018)

2.2.4. Evaluation Metrics

Evaluation metrics are critical when evaluating the effectiveness and quality of summaries produced by automatic text summarization methods. These metrics determine how the summarized text reflects the essential information for the original text data. The need for accurate evaluation stems from the overall goal of improving reliability and accuracy. This chapter first makes a nuanced distinction between intrinsic and extrinsic methods, then between manual and automated evaluations, and finally explores several well-known evaluation metrics such as BLEU, ROUGE, and others in detail.

Jones and Galliers (1995) distinguished extrinsic and intrinsic evaluation methods. Extrinsic evaluation measures the usefulness of summaries in specific application contexts such as reading comprehension or relevance assessment. In contrast, intrinsic evaluation concerns a summary's coherence and information content. A key challenge in the evaluation area is identifying and applying a robust metric to determine the adequacy of the summary. However, the notion of what constitutes a good summary is highly subjective. It depends on various factors, so existing metrics may only be appropriate for assessing some summaries (Lloret et al., 2018).

Evaluating a summary, either manually or automatically, is a significant challenge. Manual evaluation uses human feedback to assess the quality of a summary based on specific criteria such as information content, grammar, and coherence. This process is a resource and time-consuming, especially when many summaries are involved. Furthermore, despite clear evaluation guidelines, the subjective nature of manual evaluation can lead to different results depending on the evaluator (Lloret et al., 2018). Humans often disagree on selecting critical phrases from the documents used for the final summary (Jones, 2007). Nevertheless, the emergence of valuation tools such as the pyramid method, developed by Nenkova et al. (2007), offers promising approaches to address these challenges.

The pyramid method is a semi-automatic approach to evaluate the quality of summaries. A gold standard is needed for comparison with automatically generated summaries. A pyramid serves as a representation of this gold standard summary for a given set of documents, as it is used to rank the summary content. The units of comparison within the pyramid represent units of meaning called Summary Content Units (SCUs). The pyramid summarizes the perspectives of multiple authors, each producing a model summary for the documents in consideration. SCUs that recur in a more

significant number of human summaries are given a higher weight, making it easier to distinguish between critical content and less important content (Nenkova et al., 2007). The pyramid model has n levels corresponding to the number of model summaries, labeled consecutively from 1 to n . The SCUs are arranged in the pyramid based on their frequency in the model summaries. For instance, if an SCU occurs in 2 out of 4 model summaries, it is on the second level of the pyramid. With four model summaries, an SCU that appears in all summaries might be considered an essential building block, so it is assigned a weight of 4. On the other hand, an SCU that occurs only once is considered less important and is assigned a weight of 1 (Lloret et al., 2018). Figure 2.3 illustrates two SCUs at the top level and four at the following level, representing two of six ideal summaries, each comprising four SCUs (Nenkova et al., 2007).

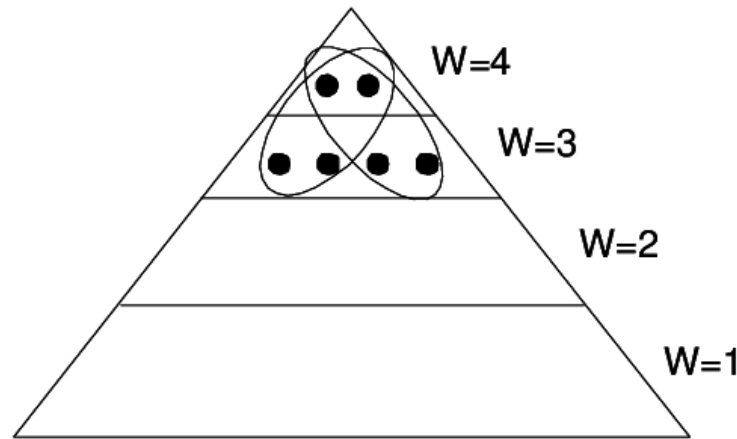


Figure 2.3.: Two out of six ideal summaries consisting of 4 SCUs (Nenkova et al., 2007)

The Bilingual Evaluation Understudy metric (BLEU), introduced by Papineni et al. (2002), is a method that was first used for quickly and inexpensively evaluating the quality of machine-translated texts compared to one or more human-produced reference translations. BLEU is designed to be language-independent and to correlate well with human judgment. The essence of the BLEU evaluation process is the comparison of n -grams in the machine-translated text with those in the reference translations. N -grams represent sequences of elements, such as words, within a text (L. Zhu et al., 2022). BLEU calculates the n -gram precision, which calculates the number of matches of n -grams between the machine translation and the human-generated reference texts. To avoid overvaluing repeated words, the count is modified by limiting the quantity of n -grams in the machine translation to the maximum number that occurs in the reference translation (Papineni et al., 2002). BLEU has been adapted to evaluate summaries and shows the

strongest correlation with overall human evaluation. The precision-based BLEU method achieves similar results to the ROUGE method, which is based on recall, in evaluating summary systems (Graham, 2015).

ROUGE is a widely recognized system for evaluating the quality of automated summaries (S. Wang et al., 2017). The name is an acronym for Recall-Oriented Understudy for Gisting Evaluation, which was introduced by C.-Y. Lin (2004). Its development was strongly influenced by the metric BLEU, which was already explained in the previous sections. ROUGE includes metrics for automatically evaluating the quality of a summary by comparing it with other human-generated summaries. These metrics count the number of matching units, such as word sequences, word pairs, and n-grams, between the machine-generated summary to be evaluated and the ideal human-generated summaries (C.-Y. Lin, 2004). N-grams represent sequences of elements, such as words, within a text (L. Zhu et al., 2022). Table 2.5 shows the different ROUGE metrics, each with different scoring requirements.

Table 2.5.: Overview of ROUGE Metrics. Adapted from C.-Y. Lin (2004).

ROUGE-N	ROUGE-N evaluates the recall of n-gram overlaps between an automatically generated summary and a collection of reference summaries created by humans. For example, ROUGE-1 evaluates the overlap of unigrams, and ROUGE-2 evaluates the overlap of bigrams.
ROUGE-L	ROUGE-L identifies word sequences that overlap, even if they are not consecutive. This metric does not take interruptions in word sequences into account and is based on the longest common subsequence. For example, “BACDG” and “BEAFCKD”, are recognized as a common sequence of “BACD”.
ROUGE-W	ROUGE-W incorporates a weighting factor to prioritize consecutive common subsequence matches. This metric extends the longest common sequence method used in ROUGE-L by considering the length of consecutive matches.
ROUGE-S	ROUGE-S uses skip-bigram statistics, where a skip-bigram is any pair of words in a sentence that allows for gaps. It measures the overlap of skip bigrams between a generated summary and reference summaries to evaluate the similarity of the content.
ROUGE-SU	ROUGE-SU overcomes a limitation of ROUGE-S, which does not recognize sentences without word pair overlaps with the references. By including the unigram count and a sentence start marker, ROUGE-SU distinguishes reversed sentences from those without word pair overlaps.

METEOR, introduced by Banerjee and Lavie (2005), is an automatic metric for evaluating machine translation and automatic text summarizations (Xiao & Carenini, 2019). It uses a method for aligning unigrams from the machine-generated translation with those from the human-generated reference translations, which makes it possible to recognize matches based on their exact spelling, their root forms, and their underlying meaning. For these matches, METEOR calculates a score by integrating unigram precision, unigram recall, and the degree of orderliness in the arrangement of words in the machine translation compared to the human-generated translation. METEOR addresses several weaknesses of BLEU, including the lack of recall, the lack of direct word matching between the translated and reference material, the reliance on advanced n-gram sequences, and the use of the geometric mean to calculate n-gram scores (Banerjee & Lavie, 2005). The

authors argue that the BLEU approach to penalizing brevity does not sufficiently offset the lack of recall. Rather than using higher-order n-grams to assess grammatical accuracy, a direct measure of grammar would be more effective. Furthermore, the indirect word matching of BLEU could lead to inaccuracies when counting N-grams. The use of geometric mean values for averaging N-grams can lead to irrelevant evaluation of individual sentences, but the use of arithmetic mean values is preferred because they reflect human evaluations more accurately (Banerjee & Lavie, 2005).

ROUGE is one of the most popular metrics in automatic text summarization, especially ROUGE-N and ROUGE-L. It is straightforward to calculate and correlates well with human judgment. A combination of metrics is often used to obtain a more comprehensive assessment, for example ROUGE and BLEU (Parida & Motlicek, 2019). In addition, METEOR is often preferred as it is intended to be an enhancement over BLEU.

2.3. Explainable Artificial Intelligence

Explainable Artificial Intelligence (XAI) marks a paradigm shift in artificial intelligence and focuses on explaining the decision-making processes of complex AI models. Deep learning methods become more powerful and complex, making it increasingly difficult to understand their behavior and the reasons for some results. Understanding these models is crucial, not only for their performance but also for building appropriate trust among users. XAI aims to make the functionality of black box models more understandable and transparent for human users (Guidotti et al., 2018).

This thesis will comprehensively analyze Explainable Artificial Intelligence (XAI) in the following sections. It begins with the historical development of XAI and then explores different types of explanations, including local and global, post-hoc, and self-explaining methods. Furthermore, various explainability techniques, such as surrogate models or feature importance, are also examined, as well as an overview of visualization techniques like raw declarative representations and heat maps. Finally, the evaluation methods for XAI are assessed, and their effectiveness and limitations are outlined.

2.3.1. Historical Overview of Explainable AI

The development of Explainable Artificial Intelligence (XAI) is a challenging process in AI, which is characterized by the increasing necessity to explain complicated algorithms. In the 1970s, in the early stage of AI, systems were created based on rule-based methods. These systems worked based on human-interpretable rules for decision-making. Despite their limited functionalities, these expert systems inherently had an explainability element. The ability to provide explanations not only increases the trustworthiness of the system but also helps non-experts to learn. The system's rules and database form a knowledge base, each representing a piece of judgmental knowledge. The reasoning process, or applying these rules, leads to the final decision. Explanations are provided by showing how the rules use user-provided information to make intermediate deductions and reach the final answer. This process ensures transparency and comprehensibility for the users (Scott et al., 1977). The decision tree is a widely used method known for its inherent explainability. As illustrated in Figure 2.4, the process starts at the top and proceeds downwards, the path taken in the decision tree showing the logical steps that lead to the final decision (Quinlan, 1990).

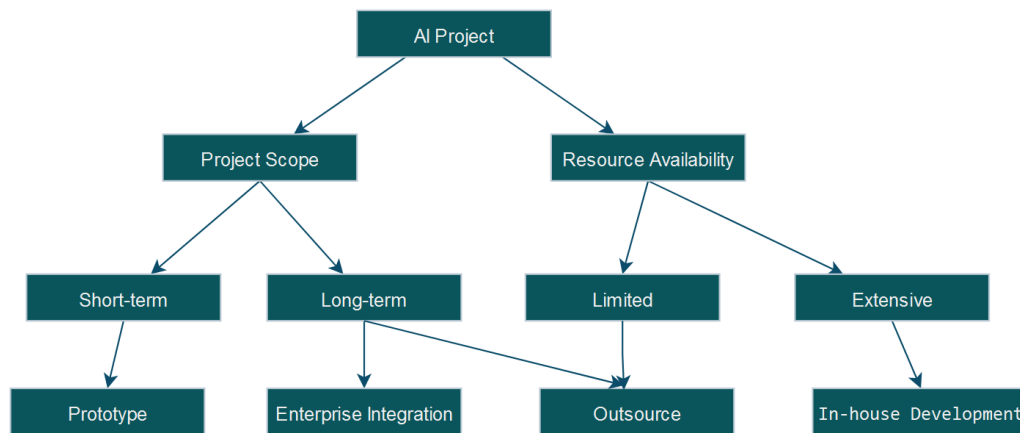


Figure 2.4.: An example of a decision tree for AI project strategy. Adapted from Quinlan (1990).

Explainable AI has recently become a significant area of study in the advanced field of deep learning. Without new explanatory methods, it is impossible to understand the results generated by modern deep neural networks. Architectures like CNN, RNN, and LSTM are designed for different problems and data inputs. However, all of these architectures are essentially black boxes, meaning that their internal processing and reasoning are not

transparent or understood by observers (Guidotti et al., 2018). A crucial step that is often overseen when building a deep learning model is explaining its logic in a clear and understandable format for humans. This explanation should highlight the biases the model has learned, allowing for a better understanding and validation of the reasons behind its decisions (Pedreschi et al., 2019). Disclosing the inner workings of a black box is essential for social acceptance and compliance. The European Union has introduced the right to an explanation for consumers with the General Data Protection Regulation (GDPR) (Goodman & Flaxman, 2017). In addition, the US Congress passed the Algorithmic Accountability Act, which requires increased control and evaluation of automated decision-making systems (MacCarthy, 2019). Explainable AI continues to be a challenging research topic, especially for highly accurate deep learning models such as transformers.

2.3.2. Explanation Types

Explanations in AI are usually categorized into four main types: global, local, self-explanatory, and post-hoc. Global explanations refer to understanding overall patterns, while local explanations focus on the reasons for specific decisions. Global explanations are crucial to gaining scientific knowledge or uncovering biases, while local explanations are necessary to justify individual decisions (Doshi-Velez & Kim, 2017).

A local explanation focuses on a single prediction instead of a global one, which reflects the overall model. Self-explanatory models are inherently transparent and understandable, such as a simple decision tree, making them accessible to most people. Post-hoc explanations, on the other hand, use auxiliary methods to explain how a model works after training (Arya et al., 2019). Table 2.6 provides a summary of these explanation categories.

Table 2.6.: Overview of Explanation Types. Adapted from Arya et al. (2019).

Global Post-Hoc	This method requires additional procedures to clarify the overall prediction logic of the model
Global Self-Explaining	The model itself is used to explain its entire predictive process
Local Post-Hoc	In this approach, additional operations are performed to clarify a single prediction after the model has reached its decision
Local Self-Explaining	The explanation for an individual prediction is derived directly from the model based on the information it produces during the prediction process

2.3.3. Explainability Techniques

There are five critical approaches to explanation, each using different mechanisms to develop the fundamental mathematical reasoning that forms the basis of the final explanation to end users.

A crucial method is feature importance, in which the relevance of the various features for the final prediction is evaluated. Several types of features are used in this method. Character-level features, for example, are often used in neural networks for natural language processing (Godin et al., 2018). Additionally, n-grams in input documents often receive importance values (Mullenbach et al., 2018). Furthermore, neural networks extract latent features, as Xie et al. (2017) described. Approaches such as the attentional mechanism developed by (Bahdanau et al., 2014) and the first derivative salience are often used to support feature importance-based explanations (Li et al., 2015).

A surrogate model serves as a simpler, more understandable model intended to clarify the functioning of a more complex model. Essentially, it is an interpretable model, trained using the results of the original, less transparent model to interpret its behavior. It is essential to recognize that there is limited theoretical certainty that these more simple surrogate models accurately represent the complexity of the original models (Adadi & Berrada, 2018). An outstanding example of this approach is the Local Interpretable Model-Agnostic Explanations (LIME) introduced by M. Ribeiro et al. (2016). LIME provides a structured approach to creating local surrogate models

that target individual instances. It addresses the more manageable task of developing a model that locally approximates the behavior of the original model for specific instances, as opposed to a global understanding that involves understanding the behavior of the original model over its entire range of inputs (M. Ribeiro et al., 2016).

Provenance-based explanations provide clarity by tracing and illustrating part or all of the process that leads to a prediction. This approach to explainability is compelling when the prediction consists of several logical steps and describes the path from the original input to the final answer (Abujabal et al., 2017). An example is QUINT, developed by Abujabal et al. (2017), which learns to generate query templates that correspond to users' questions and answers. QUINT then provides a visual representation of the reasoning sequence that converts a user's natural language question into a concrete answer. Similarly, an interpretable neural problem solver for mathematics, proposed by (Amini et al., 2019), demonstrates its reasoning by mapping problems to operations and displaying the resulting equation and the intermediate steps to solve the problem.

Declarative induction in explainable AI is about creating explanations in formats that humans can easily interpret. This includes the use of rules, decision trees, and program structures to provide clear and understandable insights (Sisodia, 2022).

Example-based explanation methods use specific instances from the dataset to show how machine learning models operate. These techniques are mainly model-independent and can make any ML model more interpretable, regardless of its underlying structure. The subtle difference from other model-independent approaches is that example-based methods focus on the behavior of a model by selecting dataset instances for interpretation rather than on manipulating features or transforming the model itself (Adadi & Berrada, 2018). These example-based explanatory methods are essentially related to the principles of nearest-neighbor-based approaches, as both methods use the fundamental principle of proximity to make decisions (Dudani, 1976). Moreover, they have been used effectively in various NLP tasks, including text classification, as shown by Croce et al., 2019.

2.3.4. Visualization Techniques

The method of explaining to the user is crucial for the effectiveness of an explainable AI system. Using the popular attention mechanism that

determines the relevance of different features, the representation could be in the form of fundamental attention values or a more accessible heat map. While both are viable, the heatmap is generally more user-friendly and has become the preferred method for visualizing attention in AI models (Danilevsky et al., 2020). This work proceeds to introduce four significant techniques for visualizing explanations in XAI.

Salience maps are often used in explainable AI to represent the significance of different elements in deep learning models. For instance, these maps can illustrate the alignment between words in an input sentence and their translation output, as shown in Figure 2.5 by Bahdanau et al. (2014). The x-axis represents the words in the original English sentence, and the y-axis in their French translation. The intensity of each pixel, which varies from black to white, indicates the weight of the contribution of each English word to the translation of a French word. An alternative method is highlighting relevant words in the input text (Schwarzenberg et al., 2019). Salience visualizations are particularly compelling because they provide intuitive, visually understandable explanations accessible to many users (Danilevsky et al., 2020).

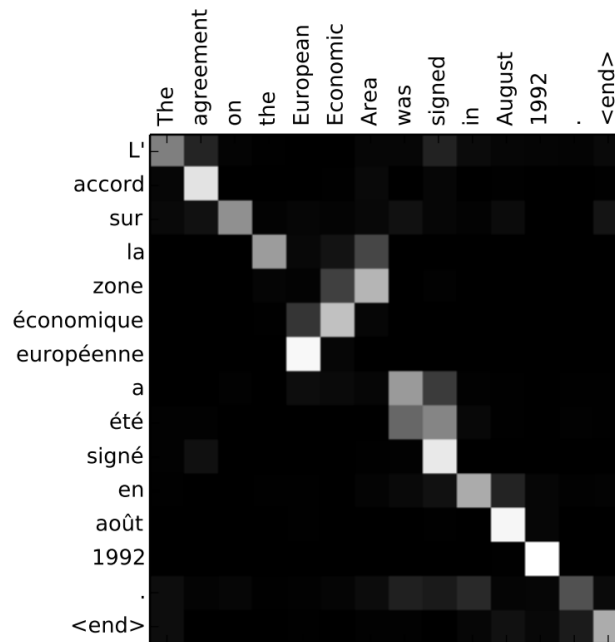


Figure 2.5.: Visualization of an attention heatmap showing the alignment between English source words and their French translations during neural machine translation. Adapted from Bahdanau et al. (2014).

Raw declarative representation is a visualization technique that directly

shows the learned structures, such as logical rules, programmatic constructs, and decision trees (Danilevsky et al., 2020). This technique assumes that users can interpret specific formats, such as reasoning trees or first-order logical rules (Pezeshkpour et al., 2019). Due to the complexity of these formats, the method is more suitable for users with a higher level of technical expertise.

Another visualization method uses raw examples to explain example-driven methods (Danilevsky et al., 2020). For instance, an NLP system developed by Croce (2019) states that the question "What is the capital of Germany?" refers to a city. It does this by drawing parallels with a similar query it has encountered, such as "What is the capital of Austria?". This comparative process is displayed to the user to illustrate the system's decision path (Croce et al., 2019).

Natural language explanations describe AI systems' reasoning in a language that humans can easily understand. This can be achieved through methods such as template-based generation (Abujabal et al., 2017). Furthermore, in approaches based on declarative induction, template-based generation methods are often used to convert complex rules and programs into easily understandable explanations and thus improve their accessibility to the general public (Reiter & Dale, 1997). Another way to obtain natural language explanations is through advanced deep learning techniques, such as training a neural language model on a unique dataset of human explanations, as shown by Rajani et al. (2019).

2.3.5. Evaluation Techniques

To effectively evaluate the utility of explanations generated by AI systems, it is vital to understand the various attributes that characterize these explanations. The various properties that are decisive for evaluating AI explanations are listed in Table 2.7.

Table 2.7.: Key properties of AI explanations. Adapted from Carvalho et al. (2019).

Accuracy	Indicates how accurately the explanation predicts the unseen data
Fidelity	Measures how closely the explanation matches the predictions of the black box model
Consistency	Examines the similarity of explanations between different models that perform the same task
Stability	Compares the consistency of explanations against slight variations of the instances for a fixed model
Comprehensibility	Refers to the degree of difficulty that people have in understanding the explanations
Certainty	Indicates the confidence level of the machine learning model in its predictions
Importance	Assesses the significance of features within the explanation
Novelty	Identifies if the instance being explained deviates significantly from the training data distribution.
Representativeness	Determines the range of instances covered by the explanation

The main evaluation techniques for explainability in AI are the comparison with ground truth, informal examination, and human evaluation.

Numerous studies evaluate the effectiveness of explanation-generation techniques by comparing them with ground truth data (Danilevsky et al., 2020). While using quantitative measures to assess the quality of explainability is a valuable approach, it is essential to ensure the integrity of the actual data and consider the possibility of multiple explanations. The metrics used for this assessment may vary depending on the task and the explanatory method. Frequently used metrics include precision, recall, and F1 scores (Carton et al., 2018). In addition, perplexity, a measure of the effectiveness of language models in predicting following words, and BLEU values are often used (Rajani et al., 2019).

Informal examination of explanations often entails extensive discussions about how the generated explanations are consistent with human reasoning (Danilevsky et al., 2020). In this process, an approach's results can be com-

pared with those of other benchmark methods. LIME, a standard frequently used in this area, is often used for this comparison (Ross et al., 2017). It can also include an independent consideration of the results of a single explanatory method (Xie et al., 2017).

A primary method for evaluating the quality of explanations is human judgment. This approach has the advantage of recognizing the diversity of compelling explanations instead of assuming a single correct explanation. Moreover, there is no need to compare the similarity of the explanations (Danilevsky et al., 2020). Crucial to this method is using multiple evaluators, reporting agreement between them, dealing with the subjectivity and variability of their responses, and comparing explanations between different methods (Sydorova et al., 2019). Furthermore, it is expected to assess explanations using a single method, such as a sentence simplifier (Dong et al., 2019).

2.4. Related Work on Explainability for Text Summarization

Before the advent of deep learning models, explainability in text summarization was more straightforward due to the use of rule-based and statistical methods that were inherently interpretable. In comparison, deep learning models for text summarization, such as those using transformers and attention mechanisms, create summaries based on complex actions that are not instantly transparent (Vaswani et al., 2017). With the growing use of these models in automatic text summarization, the need for understandable and interpretable models is becoming increasingly important. This section focuses on various studies that have examined explainability in the context of text summarization. This involves investigating techniques for visualizing model decisions, evaluating the impact of attention mechanisms, and discussing methods used to make automatic text summarization systems more trustworthy and transparent. In addition, this chapter identifies potential areas for further research and development.

Rush et al. (2015) presented an abstractive summarization model based on the neural attention mechanism, which Bahdanau introduced. This model uses a local attention approach where the input sentence influences the generation of each word in the summary. The study emphasizes the importance of attention mechanisms in the context of summaries and points to their contribution to improving the quality and relevance of the summaries produced. A heatmap illustrates the correlation between the input text and the resulting summary in this attention-based summarization system. This heatmap visually represents how the model focuses on different parts of the input as each word in the summary is created, improving the transparency and interpretability of the process (Rush et al., 2015).

The ESCA framework, introduced by H. Wang et al. (2021), is used for abstractive text summarization with a focus on controllability and explainability. It pairs an extractor, which is used to select sentences based on attributes such as informativeness, relevance, and novelty, with an abstractor, which generates summaries based on the selection. The interaction matrix of the framework highlights interactions and attributes, provides insight into the summarization process, and allows the user to influence the summary output by specifying key attributes. By integrating user preferences with advanced models such as BERT, ESCA excels at creating controlled, high-quality summaries (H. Wang et al., 2021).

Vig (2019) developed a visualization tool for the attention mechanisms in transformer models such as BERT and GPT-2, which are used in summarization tasks. This tool highlights the focus areas of the models within the input text and helps to identify biases and understand the model operations. It facilitates the identification of significant attention heads and the correlation of neuron activity with model decisions. Visual tools like these are essential for explainable AI, as they help to understand and potentially improve the decision-making processes of models based on the Transformer architecture by increasing their transparency, interpretability, and trustworthiness (Vig, 2019).

Norkute et al. (2021) examined an explanatory approach based on attention scores in a system for the automatic summarization of legal texts. The deep learning model used for this study was a sophisticated pointer generator network, which can handle abstractive text summarization tasks. The explanatory method used the model's attention mechanism to identify and highlight parts of the text that significantly impacted the summary produced. This attention mechanism focuses more on certain parts of the input data. That means the model recognizes which words in the original document are most relevant for the summary it created. The study revealed that the lawyers had to spend significantly less time reviewing the automatically generated summaries and had more trust in the results of the AI (Norkute et al., 2021).

Devlin et al. (2018) developed BERT, which is an acronym for Bidirectional Encoder Representations from Transformers. The model is designed to pre-train deep bidirectional representations by considering each layer's left and right contexts simultaneously. Hence, the pre-trained BERT model can be fine-tuned with one added output layer to develop models for various NLP tasks such as text summarization. The architecture of BERT is based on the transformer encoder, as introduced by Vaswani et al. (2017). The model comprises two main training phases: pre-training and fine-tuning. The model is trained on various tasks using unlabeled data in the pre-training phase. It is trained with a binary task to predict the following sentence to build a model that understands the sentence relationships. The BERT model starts with these pre-trained parameters for fine-tuning, which are then fine-tuned with labeled data tailored toward the NLP task (Devlin et al., 2018).

BERT has set new performance benchmarks for sentence pair regression tasks such as semantic text similarity. Nevertheless, the need to enter both sentences into the network leads to a significant computational effort. For example, to determine the most similar sentence pair from 10,000 sentences, around 50 million inference calculations are required, which amounts to approximately 65 hours for BERT. This high computational effort makes BERT less suitable for semantic similarity searches. Reimers and Gurevych (2019) developed Sentence-BERT (SBERT) to address this limitation, which is an adjustment of the pre-trained BERT network. SBERT utilizes siamese and triplet network structures to generate semantically meaningful sentence embeddings comparable by cosine similarity. This innovation reduces the time required to find the most similar sentence pair from 65 hours with BERT to only about 5 seconds with SBERT, retaining the accuracy of BERT (Reimers & Gurevych, 2019).

Meta’s Llama 2, introduced by Touvron et al. (2023), is a set of pre-trained and fine-tuned large language models. These fine-tuned models, named Llama 2-Chat, are designed for dialog scenarios and outperform other open-source models in most benchmarks. The development of Llama 2-Chat involved an initial supervised fine-tuning, followed by iterative improvement through reinforcement learning with human feedback. Llama 2 and Llama 2-Chat are available for research and commercial purposes and have been released in the parameter variants 7B, 13B, and 70B. The models were pre-trained with an improved autoregressive transformer, which uses the decoder of the transformer architecture, on 2 trillion tokens from various online data, including scientific texts, and trained on NVIDIA A100 clusters. Llama 2 doubles the maximum input sequence to 4096 tokens compared to its predecessor, which enhances its ability to summarize and understand large documents. The model also uses Grouped-Query Attention to minimize memory usage by distributing keys and values across multiple attention headers (Touvron et al., 2023).

A recognizable research gap exists regarding integrating models such as Llama 2 with Sentence-BERT. Primarily, there is a notable gap in effectively combining these models to generate concise summaries of scientific papers, followed by applying SBERT’s sentence embeddings to identify and highlight the sentences in the original text that are semantically most similar to the sentences in the summary. This provides the opportunity to develop a system that combines the language understanding of Llama 2 with the precise sentence matching of SBERT to enhance trust and understanding in automated text summaries.

2.5. Summary

The chapter begins with an overview of the development of NLP, tracing the path from Chomsky's foundational work and the ELIZA program to sophisticated transformer-based models such as GPT-4. In addition, the core methods of NLP, such as part-of-speech tagging and named-entity recognition, text classification, machine translation, and automatic text summarization, will also be addressed.

In discussing automatic text summarization, the chapter captures the essence of extractive and abstractive text summarization. It outlines how extractive methods use various statistical, discourse-based, and machine-learning techniques to identify the key sentences of the text. In contrast, abstractive summarization is a more advanced technique that uses deep learning models and attention mechanisms to create concise and coherent summaries. Introducing attention mechanisms and transformation models such as BERT, GPT series, BART, and Llama 2 revolutionized the field by providing more coherent summaries. The chapter also emphasizes the importance of arXiv, TAC, and other datasets. In addition, evaluation metrics such as ROUGE, BLEU, and METEOR are discussed, and their importance for evaluating the quality of abstracts is stressed.

The chapter shifts the focus to explainable artificial intelligence and discusses the historical transition from rule-based systems to the complex black boxes of modern deep learning models. The need for transparency and understanding in AI, especially for sophisticated models such as transformers, is underlined. Moreover, the chapter discusses different types of explanations and techniques showing how to make AI decisions easier to interpret. Furthermore, it introduces visualization techniques such as heat maps for attention mechanisms and highlighting, which provide an intuitive insight into how models process data. These techniques make complex models more straightforward to understand and increase user trust. In addition, this chapter discusses evaluation methods for XAI, focusing primarily on human judgment in determining the effectiveness and clarity of explanations.

Finally, the chapter addresses the explainability of automatic text summarization. Innovative approaches to improve the transparency and trustworthiness of automated summaries are explored. The use of attention mechanisms to track the decision process in transformation models and techniques such as Sentence-BERT to trace the source of information in summaries are discussed.

3. Requirements and Concept

In the constantly advancing field of information technology, it is more important than ever to be able to quickly capture, understand, and trust large amounts of text. This is especially true for scientific papers, where the amount of material that grows daily can be overwhelming. This work addresses this need by developing a system that condenses large scientific documents into concise summaries. In addition, the system allows users to trace each summarized sentence back to the corresponding detailed content in the original document, thereby increasing transparency and trust while showcasing Explainable AI in practice.

This chapter defines the requirements that determine the development of such a system, covering both functional and non-functional requirements, and then concludes with an exploration of the conceptual architecture. The technologies and frameworks that make this possible will be described in Chapter 4. These requirements form the basis for the design and development of the system.

3.1. Functional Requirements

Functional requirements define the expected actions of the system and, in essence, describe what the system will do. This behavior can be described as the services, tasks, or functions that the system must execute (Malan, Bredemeyer, et al., 2001). The following functional requirements apply to the system developed in this work:

- **Document Input:**
 - The system must accept scientific papers in PDF format.
 - Given a document, the system should extract the text efficiently while retaining the structure and essential formatting that is crucial for understanding the content.

- **Preprocessing:**
 - The system must clean unnecessary whitespaces and noise from the document text without disturbing the meaningful content, such as punctuation, formulas, or special symbols.
 - The system should segment the extracted text into individual sentences using NLP techniques to prepare both the summary and the sentence comparison.
- **Summarization:**
 - The system should generate concise summaries of the scientific papers that are provided as input.
- **Sentence Embedding and Semantic Matching:**
 - The system should convert all sentences from the original document and the summary into vector embeddings.
 - It should use a method to calculate the similarity between the sentence selected by the user from the summary and each sentence from the original document and identify the most relevant sentences.
 - The original sentences should be ranked based on their similarity to the selected summary sentence, and the best matches should be displayed to the user.
- **Graphical User Interface:**
 - The system should let users view the summary created via a clear, navigation-friendly interface.
 - The system allows users to select sentences from the summary to explore the source's content. After selecting a sentence from the summary, the most similar sentence from the original document is displayed.
 - The system should display errors such as unsupported document formats, extraction errors, or processing interruptions.
 - The system allows users to delete a PDF document and upload a new one without restarting it.
 - The system should display the average waiting time for generating a summary after a summary has been generated.

3.2. Non-Functional Requirements

Non-functional requirements, also known as system qualities, describe the necessary attributes of the system, including aspects such as performance, security, and maintainability. Essentially, they set the standards for how the system's various behavioral or structural aspects should be performed (Malan, Bredemeyer, et al., 2001). The following non-functional requirements apply to the system developed in this work:

- **Performance:**
 - The system must process documents and generate summaries within a reasonable time frame.
 - The system should optimize resource usage and ensure that the system uses computing resources such as memory effectively, mainly when processing large documents.
- **Reliability:**
 - The system should be reliable, functional, and accessible to the user at least 99% of the time.
 - The system should provide consistent results under similar conditions, and the quality of the summaries should not degrade over time or with increasing data volume.
- **Usability:**
 - The user interface should be intuitive and straightforward to navigate.
 - There should be a guide for the user available with instructions on using the system and interpreting the summaries and traceability results.
- **Maintainability:**
 - The system should have a clear modular structure to easily update, maintain, and extend.
 - There should be detailed technical documentation available to ensure adequate maintenance.

3.3. Conceptual Architecture

The conceptual architecture focuses on a suitable system segmentation while avoiding the complexities of interface specification and type information. Furthermore, it serves as an effective tool for explaining the architecture to non-technical stakeholders. The conceptual architecture describes the components of the system, the tasks of the individual components, and the connections between these components (Malan, Bredemeyer, et al., 2001). Figure 3.1 presents the conceptual architecture of the system developed in this work.

The conceptual architecture of the system starts with the Input Interface, through which the user can upload a scientific paper in PDF format. The document receiver is where the system initially accepts the document uploaded by the user, and the format validator ensures that the document meets the required format standards. Once the document has been checked, it enters the Preprocessing module. This module includes a text extraction component to extract text, a cleaning component to remove noise, and sentence segmentation to divide the text into individual sentences.

These sentences are then forwarded to the Summarization module, where a chunking mechanism organizes them into sections. A summarizer distills these chunks, which were previously summarized, into a final summary, that contains the most important information of the original paper. The summarized text is then forwarded to the Explainable AI module, which involves an embedding process that translates sentences into vector embeddings for similarity analysis. The sentence the user selects from the summary is compared with all sentences in the original scientific paper. A ranking system then prioritizes the sentence that is most similar in terms of semantic meaning.

Finally, the results are displayed to the user via the Graphical User Interface, which contains fields for the summarized text, the most similar sentence, and the sentence selection for capturing user interactions and triggering the similarity analysis. The similarity analysis shows the user the most similar sentence from the original document. This architecture ensures a streamlined and user-friendly process from submitting the document to receiving the summary and performing the similarity analysis, ensuring optimal performance.

3. Requirements and Concept

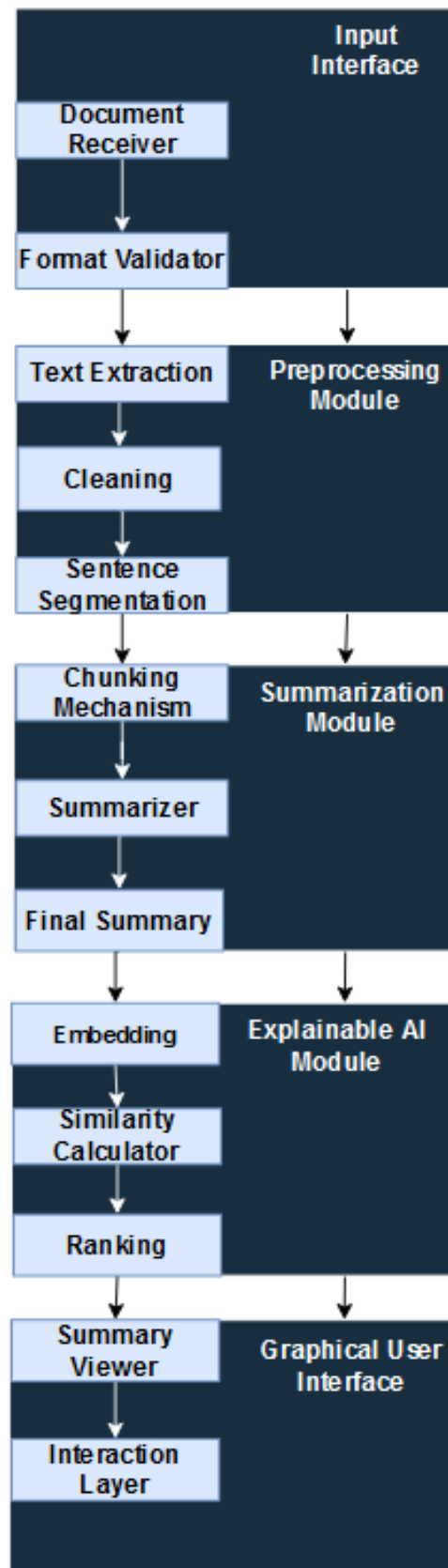


Figure 3.1.: The Conceptual Architecture of the system

3.4. Design Decisions

This section addresses the details of the design decisions. It highlights the methodical approach to meet functional and non-functional requirements to ensure a seamless transition from concept to architecture. The design decisions for the development of this system were mainly driven by the need to process and summarize large volumes of scientific texts efficiently.

The widespread use of this format in scientific literature has influenced the decision to only accept documents in PDF format. This decision simplifies the initial input phase and enables specific text extraction and pre-processing. The decision to implement the Preprocessing module results from the complexity of scientific texts. This module includes text cleaning and normalization functions that remove irrelevant spaces and noise while retaining important information. In addition, it can process various PDF encoding formats and uses a combination of regex-based patterns and NLP techniques to identify and exclude unimportant sections.

The choice of the Llama2-7B-Chat model for the Summarization module was based on its outstanding ability to handle complex linguistic structures and generate coherent summaries, as mentioned in Chapter 2. Integrating this model into the system requires a deep understanding of its capabilities and limitations, leading to optimizations that ensure superior results without exceeding resource constraints. The decision to implement a chunking mechanism was a direct consequence of the challenge of summarizing large scientific documents that exceed the maximum input sequence length of the model.

The design of the Explainable AI module was based on the latest developments in sentence embedding techniques. The choice of the all-MiniLM-L6-v2 model, a variant of the SBERT architecture, was crucial for this module. The model selection was influenced by its proven effectiveness in quickly generating high-quality sentence embeddings, which are crucial for accurate semantic similarity analysis.

The graphical user interface is developed with a focus on simplicity and functionality. Acknowledging that end users may not be technical experts, the interface was designed to be intuitive and seamlessly guide the user from uploading the document to exploring the summary. This design approach directly addresses usability requirements and ensures the system is easy to navigate.

The modular architecture of the system increases its reliability and ease of maintenance. Each component is designed to function independently, allowing for straightforward updating and maintenance. This modular design also contributes to the system's robustness and ensures that it remains operational and accessible.

3.5. Summary

This chapter describes the requirements for the system developed in this work that condenses large scientific documents into concise summaries while maintaining transparency and trust. Both functional and non-functional requirements, as well as the conceptual architecture, are discussed. The functional requirements include document input, preprocessing, summarization, semantic matching, and the graphical user interface. The non-functional requirements focus on performance, reliability, usability, and maintainability. The conceptual architecture describes the workflow from document receipt to the summary and similarity analysis presentation in a graphical user interface. This system aims to make the handling and understanding of vast volumes of scientific texts more efficient and user-friendly by tracing each summarized sentence back to the most similar sentence in the original document. The decisions for the system's design were guided by the need to efficiently process extensive scientific texts and produce accurate, trustworthy summaries. The architecture of the system has been carefully designed to be modular and scalable, offering an intuitive user interface for improved usability while ensuring high performance and reliability.

The following chapter delves into the technical details of the development process. It contains a detailed description of the steps, frameworks, and specific technologies used to meet these requirements to ensure a comprehensive understanding of how the system works and how it can be optimized for performance and accessibility.

4. Development

This chapter provides a comprehensive overview of the components of the system, including an in-depth explanation of the implementation process and the technical details of these components. This chapter builds on the previously discussed requirements and describes the steps, frameworks, and specific technologies used to fulfill these requirements in order to provide a comprehensive understanding of system functionality. Furthermore, a simplified architecture, code snippets, and screenshots are presented to enhance understanding.

4.1. Architecture

To meet the performance requirements, Google Colab Pro+¹ was initially used during development, giving access to an NVIDIA A100 Tensor Core GPU with 40 GB GPU memory², which ensures fast processing and high-quality results. In addition, Meta's large language model Llama2-7B-Chat³ was used, which requires around 30 GB of GPU memory⁴. Moreover, Gradio⁵, which can be run directly in a Google Colab notebook, is used to create a user interface for the system. Gradio is an open-source Python⁶ library that makes it simple to share interactive machine-learning programs. When the interface is started in Colab, Gradio provides a public link that can be accessed in the browser. This link is a tunnel to the web server that runs in the Colab environment and enables external access to the Gradio interface, where the user can interact with the system.

¹<https://research.google.com/colaboratory>

²<https://www.nvidia.com/en-us/data-center/a100/>

³<https://ai.meta.com/llama/>

⁴<https://docs.nvidia.com/ai-enterprise/workflows-generative-ai/o.1.0/sizing-guide.html>

⁵<https://www.gradio.app/>

⁶<https://www.python.org/>

Figure 4.1 presents the simplified architecture of the system. The system's architecture begins with the input interface, a gateway for uploading PDF documents. The Document Receiver component initiates the processing pipeline when a scientific paper is received. It is configured to ensure the system only accepts valid PDF files to avoid incorrect file formats.

After validation, the Text Extraction component parses the PDF content. This component is optimized to handle different PDF encoding formats and ensures efficient text extraction. The extracted text is then passed to the Cleaning component, part of the more extensive text extraction process. This component removes artifacts and prepares the data for further analysis. The next layer of the architecture is sentence segmentation, in which the system tokenizes the text into individual sentences. This process, crucial for the subsequent summarization phase, considers linguistic nuances such as abbreviations and complex sentence structures.

The Summarization Module then utilizes a Chunking Mechanism to split the text into several parts so that the input capacity of the model is not exceeded. The tokenizer is specially configured for the model, converting text into a processable format for the LLM. Subsequently, the Llama2-7b-Chat model creates several summaries, which are combined, and then the model creates the final concise summary. The summarized text is then passed to the Explainable AI Module, where the SBERT-based model all-MiniLM-L6-v2⁷ creates vector embeddings for sentences from the summary and the original text. By calculating the cosine similarity between these embeddings, the system can identify and display the sentences in the original document most semantically similar to those in the summary. The Hugging Place platform is a critical component of the system architecture that serves as a gateway for accessing the SBERT-based all-MiniLM-L6-v2 model and the Llama 2-7B-Chat model.

The Gradio user interface is designed for ease of use. It allows the user to interact with the system by uploading documents, viewing summaries, and selecting summary sentences to find related content in the original document. The Interaction layer is an abstract representation of the system's backend logic that processes user interactions. It ensures real-time responsiveness and updates the Gradio interface with the summary and similarity analysis results. The system's data flow has been carefully designed to ensure a seamless transition between modules. Each architectural component is designed to be loosely coupled so that individual modules can be updated independently.

⁷<https://www.sbert.net/>

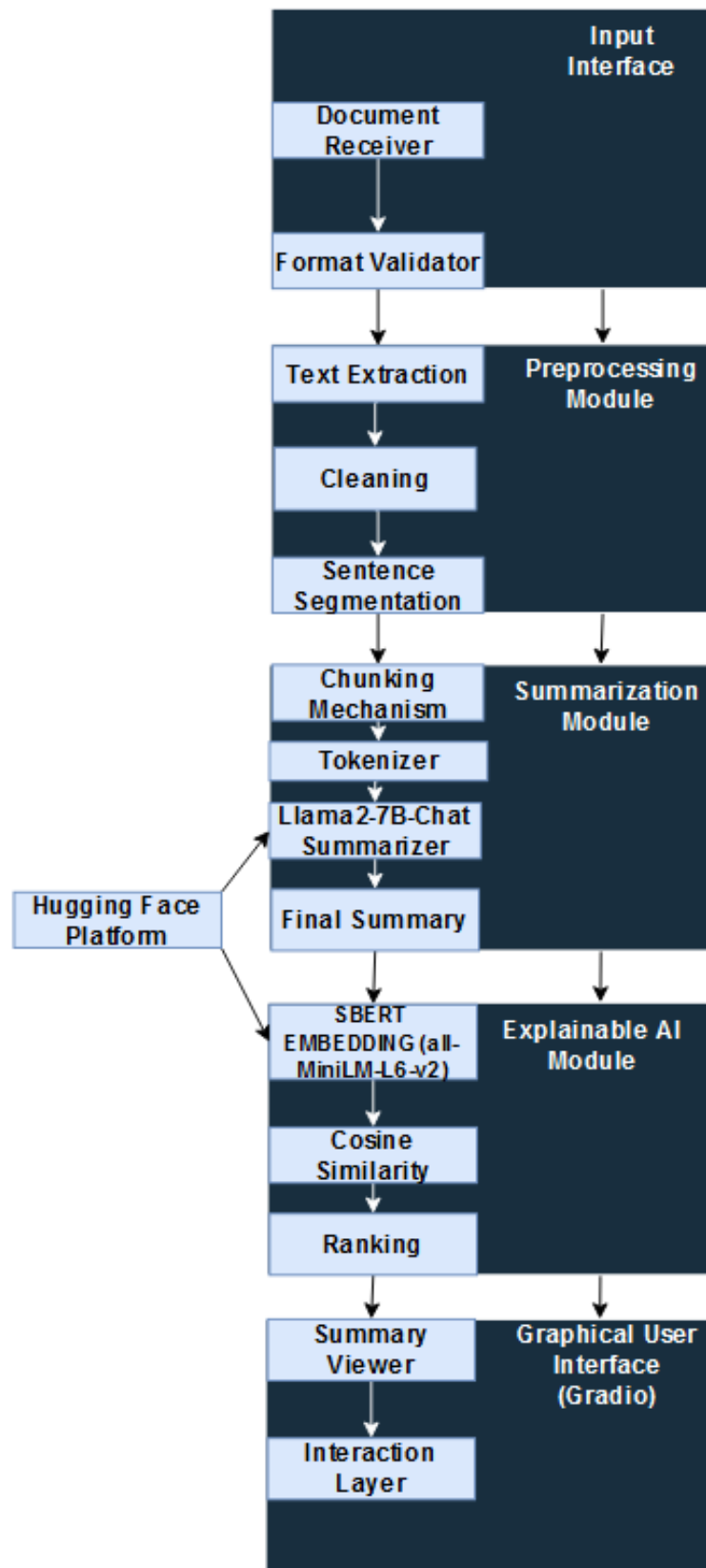


Figure 4.1.: The flowchart outlines the simplified architecture of the summarization system and highlights the core components and their connections to each other

4.2. Development Details

This section covers the development details of the previously outlined modules. It specifies the particular frameworks and technologies utilized in building the system.

4.2.1. Input Interface and Preprocessing

Figure 4.2 shows the system's input interface. It is created using Gradio, which offers a straightforward and secure method for uploading files. The system uses the file type specifications provided by the Gradio file component to verify that it is a PDF document before processing begins. Once the file has been successfully uploaded, as shown in Figure 4.3, the system uses the PDFMiner⁸ library to extract the text. The `extract_text` function takes the PDF file path as input and retrieves all the text it contains. This extracted text forms the basis for all further processing steps.

After extraction, the program starts a pre-processing phase. The program uses the `re`⁹ module to handle regular expressions and scan the extracted text carefully. The aim is to identify and exclude sections such as references or bibliographies that are generally irrelevant to the summarization process. By filtering out these sections, the program ensures that the focus remains exclusively on the relevant and informative content for the summary. In addition, the headings are marked according to a specific pattern so that they can be easily identified and processed in later phases, such as during tokenization.

In the next step, the program addresses the need to divide the text into smaller, more manageable segments. This segmentation is essential for dealing with extensive texts and is facilitated by the `RecursiveCharacterTextSplitter` class from the `langchain.text_splitter`¹⁰ module. The splitter considers parameters such as `chunk_size` and `chunk_overlap`, which determine the size of the individual text chunks and the extent of the overlap between successive chunks. These parameters are crucial as they influence the consistency of the text segments to be summarized. To avoid truncation, the `chunk_size` is 4096 characters, as this length should be less or equal to the maximum input sequence length of the Llama-2-7b-Chat model, which

⁸<https://pypi.org/project/pdfminer/>

⁹<https://docs.python.org/3/library/re.html>

¹⁰<https://python.langchain.com/>

4. Development

is 4096 tokens. The recursive basis of this splitter means that it can iteratively split the text into smaller chunks as required, ensuring that no segment exceeds the specified chunk size. The `chunk_overlap` parameter is set to 50 characters. This overlap ensures a smooth transition between successive text sections, reducing the risk of losing context or cutting off vital information at section boundaries. This overlap is significant in dense and complex texts such as scientific papers, where the continuity of ideas is crucial for an accurate summary.

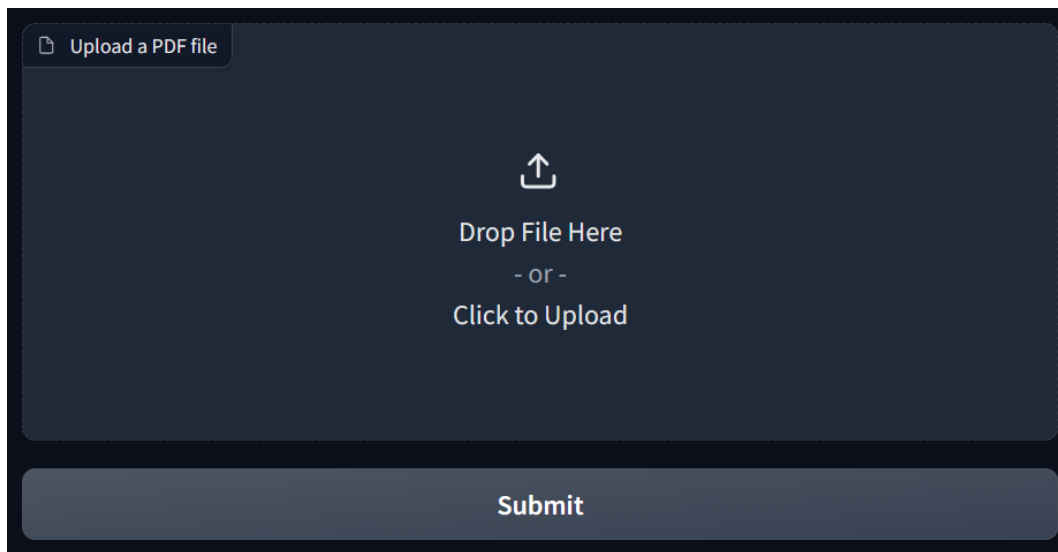


Figure 4.2.: The input interface for uploading PDF documents to the summarization system, with drag-and-drop function

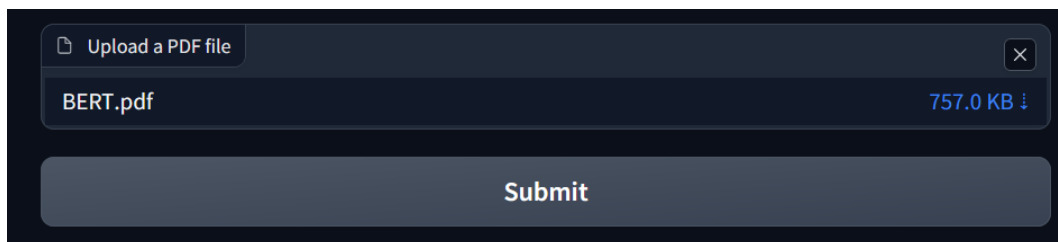


Figure 4.3.: Display of the interface after successful upload of the PDF document

4.2.2. Summarization Module

A key component in implementing the system is the setup and use of the large language model Llama2-7B-Chat, which was already introduced in Chapter 2.4. The following code snippet in Listing 4.1 illustrates the initialization process of this model. In order to use the large language model Llama2-7B-Chat, it is first necessary to obtain an access token from the official Meta AI website¹¹. This token is then used for authentication and to access the model on Hugging Face¹², a platform offering an extensive repository of pre-trained models. Furthermore, a separate token from Hugging Face is required to use the model in the Google Colab environment.

The code snippet starts by loading the necessary configuration from a JSON file that contains the essential Hugging Face token. If the token is available, the program starts to set up the Llama2-7B-Chat model. The initialization of the tokenizer, a core component of the NLP pipeline, is specifically tailored to this model and facilitates the conversion of input text into a format that the model can process. Subsequently, a text generation pipeline is set up that uses Hugging Face's transformers¹³ library. This pipeline abstracts the complexity of text processing and makes it more straightforward to interact with the model. The most critical parameters are carefully selected to optimize the performance of the pipeline. The `max_length` parameter is set to 4000, a limit that enables extensive summary creation and, at the same time, does not exceed the processing capabilities of the model. The pipeline configuration also sets `do_sample` to true and `top_k` to 5, which ensures diverse but still coherent text generation, which is crucial for natural fluency in summaries. Furthermore, `num_return_sequences` is set to 1 to ensure that the output consists of only one summary per chunk.

Finally, the `HuggingFacePipeline` class from the langchain¹⁴ framework is configured, which encapsulates the text generation pipeline. For a deterministic output, the temperature parameter is set to 0 to maintain the consistency of the summaries generated by our system. The system's design also includes an error-handling mechanism, raising an exception if the Hugging Face token is not found in the configuration.

¹¹<https://ai.meta.com/llama/>

¹²<https://huggingface.co/>

¹³<https://huggingface.co/docs/transformers/>

¹⁴<https://python.langchain.com>

4. Development

```
1 config = load_config()
2 if config and 'huggingface_token' in config:
3     model_name = "meta-llama/Llama-2-7b-chat-hf"
4     tokenizer = AutoTokenizer.from_pretrained(model_name, token
5 =config['huggingface_token'])
6     text_gen_pipeline = pipeline(
7         "text-generation",
8         model=model_name,
9         tokenizer=tokenizer,
10        token=config['huggingface_token'],
11        torch_dtype=torch.bfloat16,
12        trust_remote_code=True,
13        device_map="auto",
14        max_length=4000,
15        do_sample=True,
16        top_k=5,
17        num_return_sequences=1,
18        eos_token_id=tokenizer.eos_token_id
19    )
20    llm = HuggingFacePipeline(pipeline=text_gen_pipeline,
21                             model_kwargs={'temperature': 0})
22 else:
23     raise ValueError("Huggingface token missing from
24 configuration.")
```

Listing 4.1: Initializing the model from Hugging Face

In the system, the `generate_summary` function, as shown in Listing 4.2, is central to condensing the text chunks from the previous chapter into informative summaries. The function has two parameters: `text_chunk`, the text segment to be summarized, and `sentence_number`, which specifies the desired length of the summary in the form of sentences. This adaptability is particularly important to ensure that the summary is proportionate to the size of the text chunks, while also setting a lower limit on the length of the summary to avoid scenarios where a summary might be too short.

The `PromptTemplate`¹⁵ from the langchain framework facilitates the prompt construction process. This template formulates a structured prompt that instructs the language model to create a concise summary of a specific length. The summary task is performed by the `LLMChain`¹⁶ class, which is also part of the langchain framework. This class manages the interaction with the model by providing the prompt and processing the model output to get the summary.

¹⁵https://python.langchain.com/docs/modules/model_io/prompts/pipeline

¹⁶<https://api.python.langchain.com/en/latest/chains/langchain.chains.llm.LLMChain.html>

4. Development

```
1 def generate_summary(text_chunk, sentence_number):
2
3     template =f"""
4         Write a concise summary of the following text
5         delimited by triple backquotes.
6         The summary should be no longer than {sentence_number}
7         sentences and should concisely capture the main points.
8         '{{{text}}}'
9         SUMMARY:
10        """
11
12     prompt = PromptTemplate(template=template, input_variables
13                             =["text"])
14     llm_chain = LLMChain(prompt=prompt, llm=llm)
15
16     summary = llm_chain.run(text_chunk)
17     return summary
```

Listing 4.2: Generating a summary for a given text chunk

After the summarization, a cleaning function is used to refine these summaries further. This refinement process removes all redundant or irrelevant elements from the summaries. While the `generate_summary` function focuses on individual text sections, the system uses the `generate_final_summary` function to aggregate individual summaries and create a final, comprehensive summary that captures the essence of the entire scientific paper. This function works with a range of sentence lengths, adding an extra layer of flexibility.

Figure 4.4 shows an example of a summary created by the system. The most critical points of the BERT paper (Devlin et al., 2018) are summarized. Using the NVIDIA A100 GPU, the system takes about one minute to summarize a 16-page document like the BERT paper, including the tokenization process explained in the next chapter. After computing the first summary, the system is developed to display the average duration of the summarization process. This feature allows users to understand the time required to summarize documents of similar length and complexity in the future.

Summary

The paper introduces BERT, a new language representation model that improves upon previous pre-training methods by jointly conditioning on both left and right context in all layers. BERT is designed to be fine-tuned with minimal additional task-specific parameters, making it a powerful and versatile model. The paper argues that current pre-training techniques are limited by their unidirectional nature, and BERT addresses this limitation by using a "masked language model" pre-training objective. BERT uses a multi-layer bidirectional Transformer encoder and is pre-trained on a large corpus of text data using a combination of left-to-right and right-to-left self-attention. The model is then fine-tuned for specific downstream tasks using a small amount of task-specific data. BERT's unified architecture allows it to be easily adapted to a wide range of tasks, and its pre-training process allows it to learn contextual relationships between words in a document. The paper proposes a new pre-training task called Next Sentence Prediction (NSP) to improve the performance of downstream NLP tasks. The authors fine-tune BERT for various NLP tasks and achieve state-of-the-art results, experimenting with different pre-training tasks and model sizes. They find that using a bidirectional representation, as opposed to a left-to-right representation, improves performance on tasks like QNLI and MNLI. The results suggest that BERT is effective for a wide range of NLP tasks and that the choice of pre-training tasks and model size can have a significant impact on performance.

Figure 4.4.: Summary of the BERT paper generated by the system

4.2.3. Explainable AI Module

The Explainable AI module is a critical component in the system to make the AI's decision-making process more transparent and easier to understand. The module uses advanced natural language processing techniques to analyze the semantic similarity between a sentence in the generated summary and sentences in the original document. The most similar sentence from the original text is then identified and presented, giving the user a direct insight into the content of the summary.

The process begins with the crucial tokenization step, in which the generated summary and the original document are broken down into individual sentences. For this purpose, The Natural Language Toolkit (NLTK) library¹⁷, a Python library for NLP, is used. The main part of this step is the PunktSentenceTokenizer¹⁸ class from the NLTK library, which is known for its efficiency in recognizing sentence boundaries. When initializing this tokenizer, the PunktParameters¹⁹ class is used to extend its inherent capabilities

¹⁷<https://www.nltk.org/>

¹⁸<https://www.nltk.org/api/nltk.tokenize.PunktSentenceTokenizer.html>

¹⁹<https://docs.huihoo.com/nltk/0.9.5/api/nltk.tokenize.punkt.PunktParameters-class.html>

by explicitly adding common abbreviations such as "et al", "etc" and "dr". This enhancement is critical because it allows the tokenizer to recognize these abbreviations as non-terminating entities, preventing the misidentification of sentence endings. This addition adds to the existing capabilities of the tokenizer in dealing with common sentence punctuation, such as periods, which are already managed by the library's default configuration.

After tokenization, the sentences undergo semantic analysis using a pre-trained model from the SentenceTransformers²⁰ framework. This framework provides several models, all hosted on Hugging Face. The models are based on the SBERT model, which was already introduced in Chapter 2.4. This model is particularly suitable for generating embeddings that capture the semantic essence of sentences. The all-MiniLM-L6-v2²¹ model was selected for developing this module, which is known for its optimum ratio between performance and speed. Trained on over a billion training pairs dataset, this model is five times faster than its counterparts while still delivering high-quality sentence embeddings. The model and its tokenizer are loaded and, if available, configured to use GPU acceleration to ensure optimal performance.

In Listing 4.3, the `find_most_similar_sentence` function is shown. This function is important for improving transparency, as it links the summarized content with the original scientific work. The code begins with pre-processing the original document and tokenizing it into individual sentences that are clearly delimited for semantic analysis. The key operation in this function begins with encoding both the selected sentence from the summary and all sentences from the scientific paper. Once the sentences are encoded, they are input into the sentence-transformer model, which generates vector representations for each sentence. Mean pooling is applied in order to standardize these vectors for comparison. This process condenses the multiple dimensions of each vector into a single embedding that comprises the entire semantic meaning of the sentence. Afterward, the cosine similarity scores are calculated. The semantic distance between embedding the clicked sentence from the summary and the embedding of the individual sentences from the scientific paper is measured. The sentence with the highest cosine similarity score is then identified as the most similar.

²⁰<https://www.sbert.net/>

²¹https://www.sbert.net/docs/pretrained_models.html

4. Development

```
1 def find_most_similar_sentence(clicked_sentence,
2   scientific_paper):
3     preprocessed_text = clean_source_text(scientific_paper)
4     source_sentences = tokenize_sentences(preprocessed_text)
5
6     encoded_clicked = tokenizer(clicked_sentence, padding=True,
7     truncation=True, return_tensors='pt').to(device)
8     encoded_sources = tokenizer(source_sentences, padding=True,
9     truncation=True, return_tensors='pt').to(device)
10
11     clicked_embedding = encode_and_embed(clicked_sentence)
12     source_embeddings = encode_and_embed(source_sentences)
13
14     clicked_embedding = F.normalize(clicked_embedding, p=2, dim
15     =1)
16     source_embeddings = F.normalize(source_embeddings, p=2, dim
17     =1)
18
19     cos_scores = torch.mm(clicked_embedding, source_embeddings.
20     T).squeeze(0)
21     most_similar_idx = torch.argmax(cos_scores).item()
22
23     return source_sentences[most_similar_idx]
```

Listing 4.3: Identifying the most similar sentence

Figure 4.5 shows the Explainable AI module of the system in practice. The user has selected a sentence from the summary of the BERT paper, and the system has determined and displayed the most similar sentence from the original document within a second. This module is crucial for the user to understand better how the content of the summary correlates with the detailed information in the scientific paper, increasing the transparency and trustworthiness of the summary produced by the AI.

4. Development

Upload a PDF file

BERT.pdf

757.0 KB ↓

X

Submit

Most similar sentence in the original document

Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

Select a sentence from the summary

☒ The paper introduces BERT, a new language representation model that improves upon previous pre-training methods by jointly conditioning on both left and right context in all layers.

☐ BERT is designed to be fine-tuned with minimal additional task-specific parameters, making it a powerful and versatile model.

☐ The paper argues that current pre-training techniques are limited by their unidirectional nature, and BERT addresses this limitation by using a "masked language model" pre-training objective.

Figure 4.5.: Displaying the most similar sentence from the original BERT paper based on the user-selected summary sentence

4.2.4. Graphical User Interface

Figure 4.6 provides an overview of the original design concept for the system's graphical user interface. This design laid the foundation for the developed intuitive and user-friendly interface.

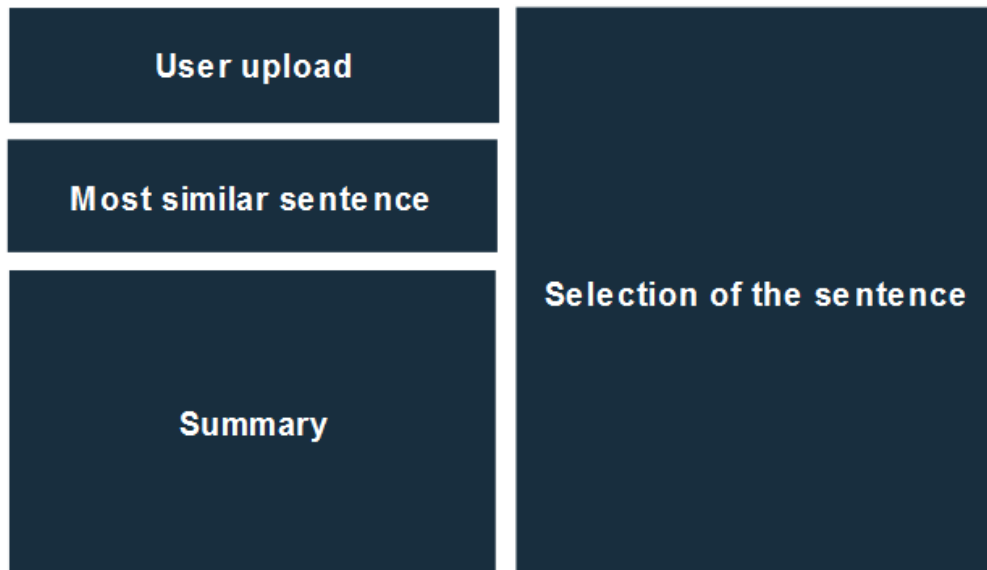


Figure 4.6.: Overview of the design concept for the system's GUI

Figure 4.7 illustrates the implementation of the system's graphical user interface, using the BERT paper again as input to demonstrate the entire process. The interface developed with Gradio is distinguished by a clear layout focusing on user interaction and the system's core functions. The top left-hand side of the user interface is dedicated to the user upload area. In this user-friendly area, documents can be easily uploaded, followed by the "Submit" button. Once a document has been submitted, it is processed as described previously. The resulting summary is displayed in the "Summary" box. In addition to the summary, the "Select a sentence from the summary" area allows users to interact with the summarized content. Users can select specific sentences from the summary. When selecting, the system dynamically determines the most similar sentence from the original document. It displays it in the "Most similar sentence in original document" field on the user interface's left side, increasing transparency and trust.

4. Development

Upload a PDF file

BERT.pdf

757.0 KB

Submit

Most similar sentence in the original document

Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pretrain deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.

Summary

The paper introduces BERT, a new language representation model that improves upon previous pre-training methods by jointly conditioning on both left and right context in all layers. BERT is designed to be fine-tuned with minimal additional task-specific parameters, making it a powerful and versatile model. The paper argues that current pre-training techniques are limited by their unidirectional nature, and BERT addresses this limitation by using a "masked language model" pre-training objective. BERT uses a multi-layer bidirectional Transformer encoder and is pre-trained on a large corpus of text data using a combination of left-to-right and right-to-left self-attention. The model is then fine-tuned for specific downstream tasks using a small amount of task-specific data. BERT's unified architecture allows it to be easily adapted to a wide range of tasks, and its pre-training process allows it to learn contextual relationships between words in a document. The paper proposes a new pre-training task called Next Sentence Prediction (NSP) to improve the performance of downstream NLP tasks. The authors fine-tune BERT for various NLP tasks and achieve state-of-the-art results, experimenting with different pre-training tasks and model sizes. They find that using a bidirectional representation, as opposed to a left-to-right representation, improves performance on tasks like QNLI and MNLI. The results suggest that BERT is effective for a wide range of NLP tasks and that the choice of pre-training tasks and model size can have a significant impact on performance.

Select a sentence from the summary

☒ The paper introduces BERT, a new language representation model that improves upon previous pre-training methods by jointly conditioning on both left and right context in all layers.

☐ BERT is designed to be fine-tuned with minimal additional task-specific parameters, making it a powerful and versatile model.

☐ The paper argues that current pre-training techniques are limited by their unidirectional nature, and BERT addresses this limitation by using a "masked language model" pre-training objective.

☐ BERT uses a multi-layer bidirectional Transformer encoder and is pre-trained on a large corpus of text data using a combination of left-to-right and right-to-left self-attention.

☐ The model is then fine-tuned for specific downstream tasks using a small amount of task-specific data.

☐ BERT's unified architecture allows it to be easily adapted to a wide range of tasks, and its pre-training process allows it to learn contextual relationships between words in a document.

☐ The paper proposes a new pre-training task called Next Sentence Prediction (NSP) to improve the performance of downstream NLP tasks.

☐ The authors fine-tune BERT for various NLP tasks and achieve state-of-the-art results, experimenting with different pre-training tasks and model sizes.

☐ They find that using a bidirectional representation, as opposed to a left-to-right representation, improves performance on tasks like QNLI and MNLI.

☐ The results suggest that BERT is effective for a wide range of NLP tasks and that the choice of pre-training tasks and model size can have a significant impact on performance.

Figure 4.7.: Graphical User Interface of the system

4.3. Summary

This chapter provides a detailed insight into the system's development process. It covers various aspects of system development, from the simplified architecture to the implementation of specific modules and the graphical user interface.

The chapter begins with a detailed discussion of the system architecture, focusing on the system's performance. Key components such as Google Colab Pro+ and the NVIDIA A100 Tensor Core GPU are highlighted to ensure fast processing and high-quality results. The chapter also introduces using Meta's large language model, Llama2-7B-Chat, and Gradio to create a user-friendly interface. The architecture is presented in a simplified diagram that outlines the entire process. The Summarization module and the Explainable AI module are the main modules. This chapter shows how they work together to create concise summaries and link them semantically to the original text.

This includes using a RecursiveCharacterTextSplitter to manage large chunks of text and a detailed explanation of the summarization process using the Llama2-7B-Chat model. This chapter explains in detail the initialization of the model, the generation of summaries, and the final aggregation of these summaries to capture the essence of the entire scientific document. The process of analyzing the semantic similarity between the summary and the original document is explained in detail in this module, with explanations of tokenization, the use of the model all-MiniLM-L6-v2 from the SentenceTransformers framework, and the calculation of cosine similarity values.

Finally, this chapter introduces the system's graphical user interface. The graphical user interface makes it easy to upload documents, view summaries, and interact with the summarized content to find related original texts, increasing transparency and user trust.

5. Evaluation

This chapter addresses the evaluation of the developed AI summary system, focusing on two central studies that provide insights into its functionality. The motivation is to assess the ability of the system to create comprehensible summaries from complex scientific papers and evaluate the system's explanatory function. The main goal of this evaluation is to assess the quality of the summaries produced by the AI system and to evaluate the transparency and impact of this process.

The chapter initially focuses on the Summarization Quality Study (Section 5.1), in which the authors analyze the quality of the generated summaries in terms of their accuracy, coherence, and completeness. Afterward, the Explainability Study (Section 5.2) is carried out to examine how well the system's explanations support users' understanding and trust. Both studies follow a similar sub-chapter structure. The study design (Sections 5.1.1 and 5.2.1), the setting (Sections 5.1.2 and 5.2.2), the instruments (Sections 5.1.3 and 5.2.3) used, and the method applied are described for each study. In addition, the results (Sections 5.1.4 and 5.2.4) are discussed in detail at the end.

5.1. Summarization Quality Study

5.1.1. Study Design

The AI Summarization Study evaluates the quality of the summaries produced by the AI, particularly the LLM Llama2-7b-Chat. The aim is to evaluate the summaries' accuracy, coherence, and completeness in relation to the original scientific texts they represent. This will examine how well the system captures the key points and presents them in a way that is both informative and accessible to the user. The user study aims to answer the following research question:

- **RQ1:** How do authors rate the quality of the summaries produced by the AI system in terms of their accuracy, coherence, and overall usefulness for understanding the original scientific literature?

5.1.2. Setting and Instruments

The developed system generated 20 summaries of scientific papers published in the Journal of University Computer Science (J.UCS) for this study. Six papers, each from Issue 1 Volume 30 and Issue 12 Volume 29, were used to generate the summaries. In addition, eight papers from Issue 11 Volume 29 were used. The study was sent to all participating authors of the 20 papers, totaling 68 people. The LLM Llama2-7B-Chat with an NVIDIA A100 Tensor Core GPU was used to generate the summaries, the exact process was already mentioned in Chapter 4.1.

The study was conducted in a virtual setting, using digital tools to facilitate the research process and data collection. Participants, the authors of the original papers, were invited to rate the summaries via an online platform to ensure a broad and diverse response base without geographical restrictions. The study utilizes Google Form¹ as the primary instrument to create an online questionnaire. The questionnaire contains a series of Likert scale questions designed to quantitatively measure authors' assessments of the summaries produced by the AI on various aspects such as accuracy, coherence, and completeness. Furthermore, the questionnaire incorporates open-ended questions, which aim to gather qualitative insights into these summaries' perceived strengths and weaknesses, providing a more nuanced understanding of the AI's performance. The instrument Google Sheets² is used to analyze the collected data, allowing for efficient organization and visualization of the responses, facilitating a comprehensive assessment of the summary's quality. In order to uniquely identify the participants, the first field asks for the DOI of the scientific paper.

The following demographic questions are asked in the Google Form survey at the beginning:

- **Q1.1:** What is your age range? (Under 18, 18-24, 25-34, 35-44, 45-54, 55-64, 65 or older)
- **Q1.2:** Which of the following best describes your gender? (Female, Male, Non-binary, Prefer not to say)

¹<https://www.google.com/forms/about/>

²<https://www.google.com/sheets/about/>

- **Q1.3:** What is the highest degree or level of education you have completed? (High school graduate, Bachelor's degree, Master's degree, Doctorate)
- **Q1.4:** What is your profession? (Student, Researcher, Professor, Teacher, Management, Engineer, Other)
- **Q1.5:** How would you rate your experience with Artificial Intelligence (AI) tools? (from 1 No Experience to 5 Highly experienced)

The following aspects of the AI-generated summaries on a scale from 1 (Very Low) to 5 (Very High) are rated:

- **Q1.6:** How would you rate the accuracy with which the summary represents the content of your paper?
- **Q1.7:** How well does the summary highlight the key contributions of your research?
- **Q1.8:** How would you rate the coherence of the AI-generated summary? (Logical progression of ideas, Clarity of connections, Consistency of tone and style)
- **Q1.9:** How would you rate your overall satisfaction with the AI-generated summary of your work?
- **Q1.10:** Is the length of the summary appropriate? (On a scale from 1 (Too Short) to 5 (Too Long))

Open questions in the survey should provide information on the following:

- **Q1.11:** If there are aspects that have not been mentioned in the summary, what are they?
- **Q1.12:** If there is content in the summary that can be omitted, what is it?

5.1.3. Procedure

The procedure involves providing authors with AI-generated summaries of their papers from the Journal of Universal Computer Science (J.UCS) and asking them to evaluate these summaries using the structured questionnaire. This assessment covers a wide range of criteria, including the accuracy with which the summary reflects the original work, the coherence of the narrative, and the coverage of key points. This process directly compares the AI summary with the author's expert knowledge of their work.

The corresponding authors were contacted individually by email and asked to complete the survey. In the email, the reasons for conducting the study were explained. The email included the estimated time frame for completing the survey, the title of the authors' summarized paper, and the DOI. The generated summary and a link to the online survey were at the end. An email was also sent to the co-authors. The authors' privacy and data confidentiality were paramount when conducting this study. Before the data was collected, the participants were given a data protection declaration in which they were assured that the survey was conducted exclusively for scientific research purposes. The statement clarified that all responses would be anonymized and only used in aggregate form for analysis to ensure that no individual responses would be identifiable in a published work.

5.1.4. Results and Discussion

A total of five papers by five authors were evaluated, meaning a quarter of the summaries generated were assessed. The demographic data of the participants is described in the following. Figure 5.1 shows the age range of the authors, which indicates that the largest segment at 40% (2 participants) is individuals aged 25-34 (Q1.1). There is also one author in each of the age ranges 18-24, 45-54, and 65 or older. Figure 5.2 illustrates the participants' gender, indicating 60% (3 participants) men and 40% (2 participants) women (Q1.2). Figure 5.3 depicts the highest educational degree of the authors, showing that 60% (3 participants) have a Doctorate and 40% (2 participants) have a Master's degree (Q1.3). The profession of the authors is presented in Figure 5.4. The professions of the participants included 20% (1 participant) students, 60% (3 participants) researchers, and 20% (1 participant) engineers (Q1.4). The majority is experienced using AI tools, as shown in Figure 5.5 (Q1.5).

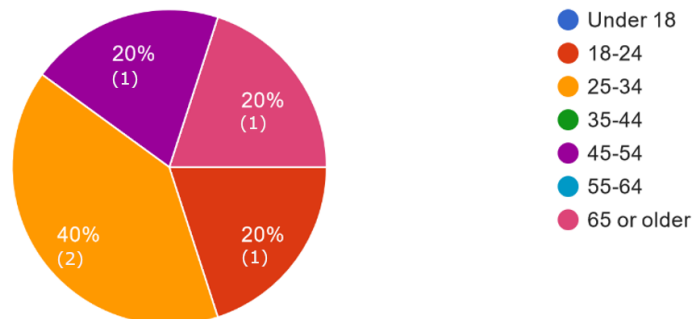


Figure 5.1.: Age range of authors (Q1.1)

5. Evaluation

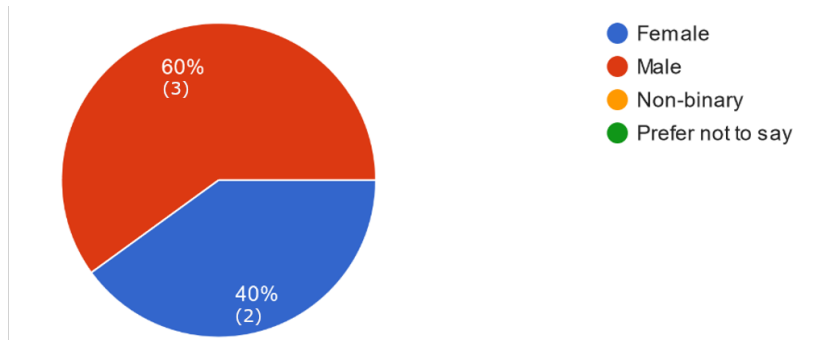


Figure 5.2.: Gender of authors (Q1.2)

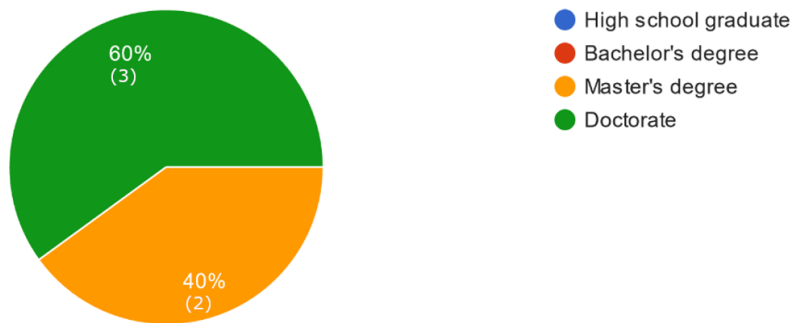


Figure 5.3.: Highest degree or level of education (Q1.3)

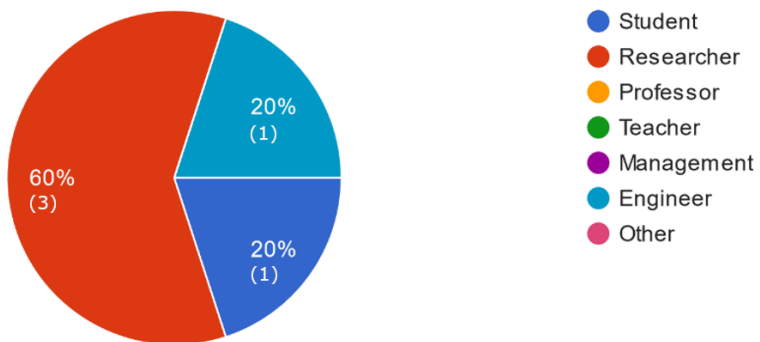


Figure 5.4.: Profession of authors (Q1.4)

5. Evaluation

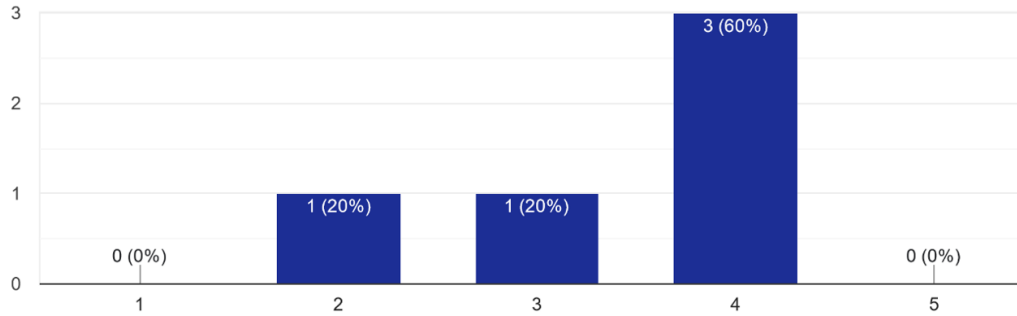


Figure 5.5.: Experience with Artificial Intelligence (AI) tools (Q1.5)

The following section discusses different criteria for evaluating the generated summaries. On average, the accuracy of the AI summaries is rated very highly. Four authors rated this aspect with 4 out of 5, and one even with a perfect 5 (see Figure 5.6, Q1.6). Most responses (60%; 3 participants) rated the AI's ability to highlight the critical contributions of research at 5 out of 5, indicating a solid performance (see Figure 5.7, Q1.7). However, two responses with lower scores of 2 and 3 indicated some deviation in the AI's performance. The authors rated the coherence of the summaries produced by the AI differently, with scores ranging from 2 to 5, as shown in Figure 5.8 (Q1.8). Two authors rated coherence as high (scores 5 and 4), while three authors rated the coherence lower (scores 2 and 3), indicating that the performance of the AI system is inconsistent across papers. The overall satisfaction with the summaries produced by the AI shows that 60% (3 participants) of respondents rated 4, indicating a high satisfaction level (see Figure 5.9, Q1.9). In addition, 40% (2 participants) of respondents rated the value 3, which suggests a medium level of satisfaction. The feedback on the length of the summaries created by the AI shows that the majority consider the summaries to be appropriate, as depicted in Figure 5.10 with a score of 3 (Q1.10). Two participants (20%) rated the length as 4 and 5, which indicates that the summary was too long for their work. This could be due to several reasons, such as the complexity of the work or the amount of information the author considers essential. Table 5.1 shows an overview of the authors' responses to the quantitative evaluation criteria.

5. Evaluation

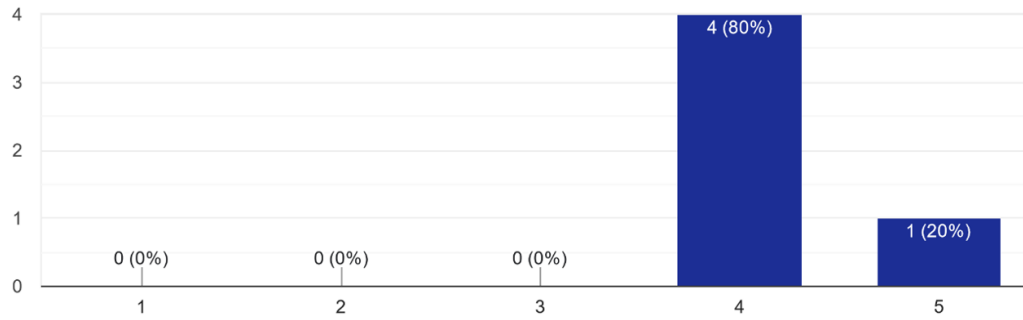


Figure 5.6.: Accuracy with which the summary reflects the content (Q1.6)

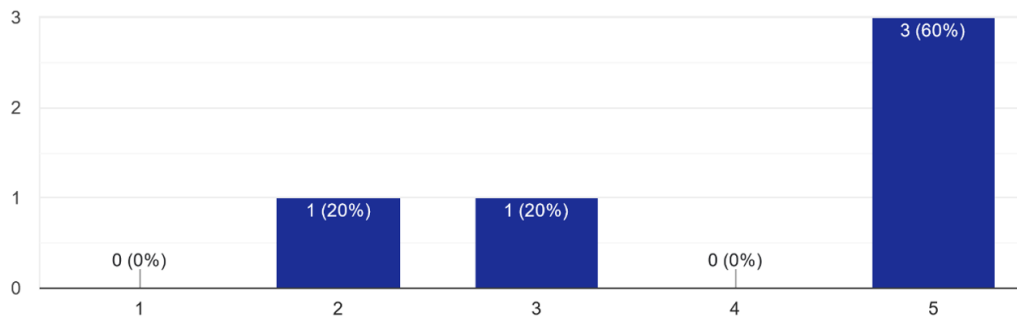


Figure 5.7.: To what extent the summary highlights the key contributions (Q1.7)

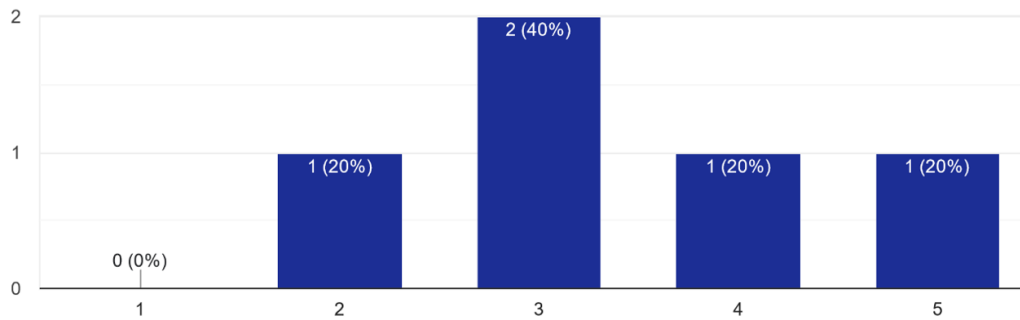


Figure 5.8.: Coherence of the AI-generated summary (Q1.8)

5. Evaluation

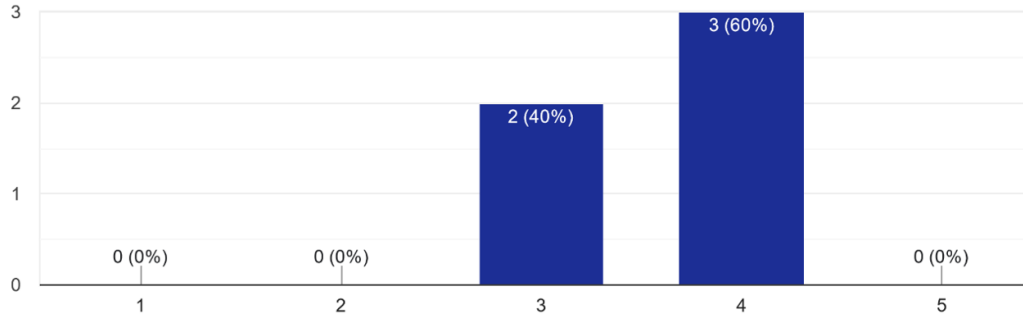


Figure 5.9.: Overall satisfaction with the summary (Q1.9)

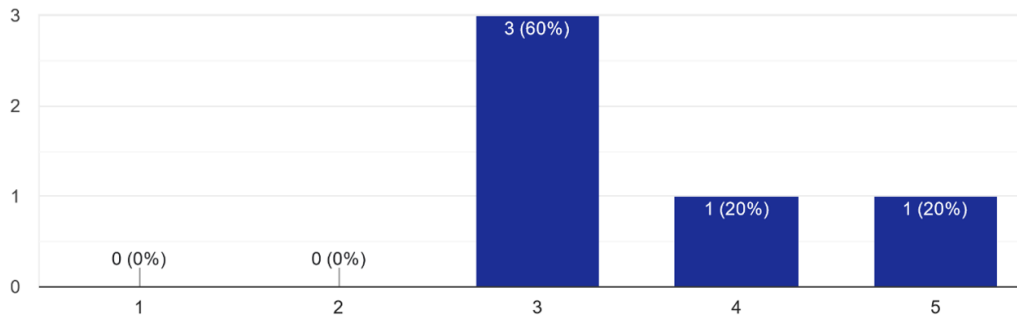


Figure 5.10.: Appropriate length of the summary (Q1.10)

Table 5.1.: Overview of the authors' responses to the quantitative evaluation criteria.

ID	Question	Mean	STD
Q1.6	How would you rate the accuracy with which the summary represents the content of your paper?	4.20	0.40
Q1.7	How well does the summary highlight the key contributions of your research?	4.00	1.26
Q1.8	How would you rate the coherence of the AI-generated summary? (Logical progression of ideas, Clarity of connections, Consistency of tone and style)	3.40	1.02
Q1.9	How would you rate your overall satisfaction with the AI-generated summary of your work?	3.60	0.49
Q1.10	Is the length of the summary appropriate? (3 = appropriate)	3.60	0.80

The answers to the open questions (Q1.11, Q1.12) highlighted some areas for improvement. One response suggested that the acknowledgments for

projects supporting the research could be omitted from the summary (*“Acknowledgments to the projects that have supported the content of the article”*). Another participant noted that the summary contained too many repetitions of the same vital aspects (*“The summary contains many repetitions of the same key aspects, resulting in more sentences than necessary.”*). This feedback shows that the AI system needs to better distinguish between core content and more secondary information. It also tends to overemphasize specific points, leading to redundancy and increasing the summary length.

The AI summary system is generally effective in producing coherent and accurate summaries highlighting the most important research contributions. However, the variations in some responses, particularly in highlighting key contributions, indicate that there is still room for improvement. The different demographic characteristics and experience with AI tools among the respondents show that this is a diverse sample. Nevertheless, the number of participants would need to be increased to evaluate the AI summary system better. Given the professional background of the authors, the positive reviews suggest that the AI summaries are likely to meet the high standards of an academic target group.

5.2. Explainability Study

5.2.1. Study Design

This study aims to critically evaluate the transparency and interpretability of the AI summary system. This aspect of the study concerns the ability of the system to make its functioning more understandable to users, which is essential for trust and practical application in academic and professional environments. This study aims to answer the following research question:

- **RQ2:** How effectively does the AI summary system provide transparency in its summary process, and how does this transparency impact user trust?

5.2.2. Setting and Instruments

Participants will receive screenshots with the results of the AI system, including the summaries generated by the AI, to enable a concrete evaluation

of the explainability of the system. In addition, selected sentences from the summaries are shown together with the most similar sentences from the original scientific paper calculated by the system. These screenshots are shown in Appendix A and serve as a direct visual reference for participants, aiding in their assessment of the system's transparency and the quality of its summarization. The LLM Llama2-7B-Chat with an NVIDIA A100 Tensor Core GPU was used to generate the summaries, and the SBERT-based model all-MiniLM-L6-v2 was utilized for the similarity outputs. The exact workflow was already mentioned in Chapter 4.1. Similar to the first study, this study also uses a Google Forms questionnaire with Likert scale questions to measure the clarity of the AI summarization process and the trustworthiness of the summaries. Moreover, Google Sheets analyzes and presents the collected data in a suitable format. The following demographic questions are asked in the Google Form survey at the beginning:

- **Q2.1:** What is your age range? (Under 18, 18-24, 25-34, 35-44, 45-54, 55-64, 65 or older)
- **Q2.2:** Which of the following best describes your gender? (Female, Male, Non-binary, Prefer not to say)
- **Q2.3:** What is the highest degree or level of education you have completed? (High school graduate, Bachelor's degree, Master's degree, Doctorate)
- **Q2.4:** What is your profession? (Student, Researcher, Professor, Teacher, Management, Engineer, Other)
- **Q2.5:** How would you rate your experience with Artificial Intelligence (AI) tools? (from 1 No Experience to 5 Highly Experienced)

Participants use the questionnaire to rate on a scale from 1 (Very Low) to 5 (Very High) the following aspects:

- **Q2.6:** How would you rate the clarity of the AI-generated summaries?
- **Q2.7:** To what extent do you trust the summaries produced by the AI system to be accurate and reliable?
- **Q2.8:** How effective do you think the system is in allowing users to trace the summary content back to the original document?
- **Q2.9:** How would you rate the overall quality of the summaries in terms of their coherence? (Logical progression of ideas, Clarity of connections, Consistency of tone and style)
- **Q2.10:** To what extent does the transparency of the system contribute to general trust in the system?
- **Q2.11:** To what extent do you believe that the explainability function provided by the AI system supports you in your work when interacting with the summaries?

The open questions in the questionnaire are intended to obtain detailed feedback on the following points:

- **Q2.12:** What improvements or additional features would enhance the system's explainability for you?
- **Q2.13:** What are the strengths and weaknesses you perceive in the AI summarization system?

5.2.3. Procedure

The study should include a diverse group of at least 30 participants, comprising researchers, Master's students, Bachelor's students, and high school graduates, ensuring a wide range of perspectives and expertise. Participants were invited to participate in the study by email and social media channels, which contained a brief description, the estimated time frame for completing the survey, and the online survey link at the email's end. When conducting this study, the privacy of the participants and the confidentiality of the data was of primary importance. Before the data was collected, the participants were given a data protection declaration in which they were assured that the survey was conducted exclusively for scientific research purposes. The statement clarified that all responses would be anonymized and only used in aggregate form for analysis to ensure no individual responses would be identifiable in a published work.

At the beginning of the study in Google Forms, the user is given an explanation of how the system works with the help of images and an easy-to-understand description. Furthermore, there is a YouTube video³ provided that shows how the system works. The study is carefully organized to provide participants with a coherent and engaging assessment task. Through the presentation of AI-generated summaries and their explanations, participants are immersed in a practical assessment scenario. Participants are tasked with reviewing the summaries and their explanations produced by the AI and evaluating them against various criteria such as clarity, trustworthiness, and effectiveness in linking the summaries to the original texts. This practical assessment is designed to provide robust and meaningful feedback.

³<https://www.youtube.com/watch?v=cS6KjgvsGJg>

5.2.4. Results and Discussion

A total of 34 participants were involved in the study. The demographic data of the participants is described in the following. Figure 5.11 (Q2.1) illustrates the age range of the participants, which shows that the majority of participants were between 25 and 34 years old (50%; 17 participants), followed by the 18-24 age group (47.1%; 16 participants). Only one participant (2.9%) is between 45 and 54 years old, which indicates a range of perspectives, especially from younger adults. Figure 5.12 (Q2.2) shows the participants' gender, indicating 61.8% (21 participants) men and 35.3% (12 participants) women. One participant (2.9%) did not want to specify their gender. Figure 5.13 (Q2.3) presents the highest educational degree of the participants, showing that the majority of participants had completed a Bachelor's degree (58.8%; 20 participants), followed by Master's degree holders (17.6%; 6 participants) and high school graduates (17.6%; 6 participants). In addition, two participants (5.9%) have a Doctorate. The profession of the participants is depicted in Figure 5.14 (Q2.4). The participants' professional backgrounds varied, with students forming the largest group (52.9%; 18 participants). Engineers were the next largest group (20.6%; 7 participants). Teachers, managers, and researchers were also represented in smaller numbers. A substantial proportion of participants (44.1%; 15 participants) rated their experience with AI tools at level 3 on a scale of 1 (no experience) to 5 (very experienced), as shown in Figure 5.15 (Q2.5). This indicates a medium level of familiarity with AI technologies.

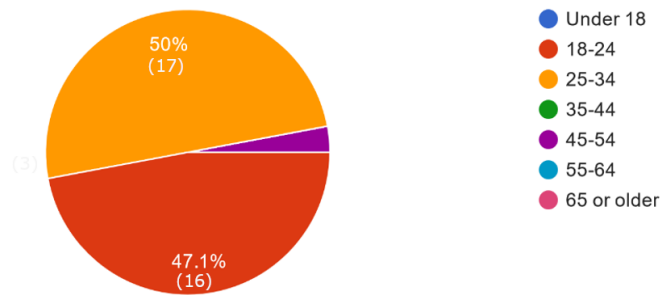


Figure 5.11.: Age range of participants (Q2.1)

5. Evaluation

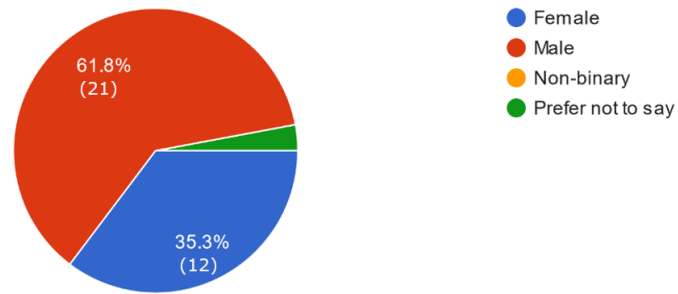


Figure 5.12.: Gender of participants (Q2.2)

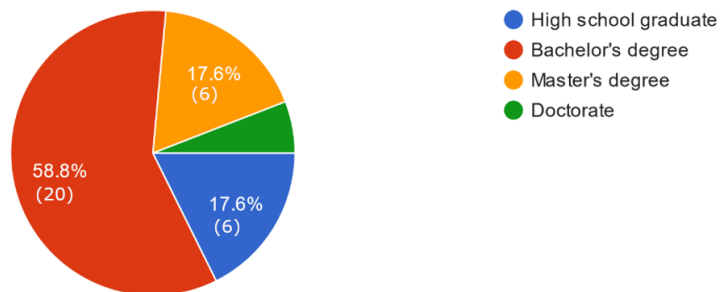


Figure 5.13.: Highest degree or level of education (Q2.3)

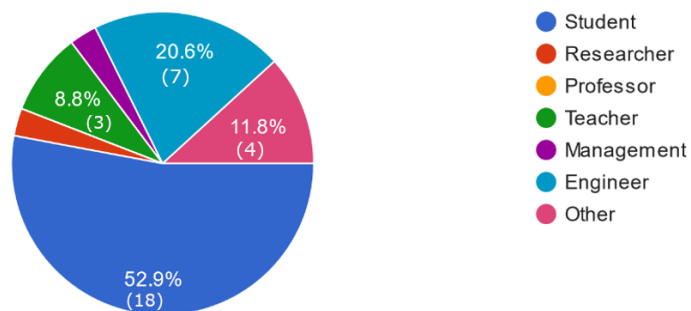


Figure 5.14.: Profession of participants (Q2.4)

5. Evaluation

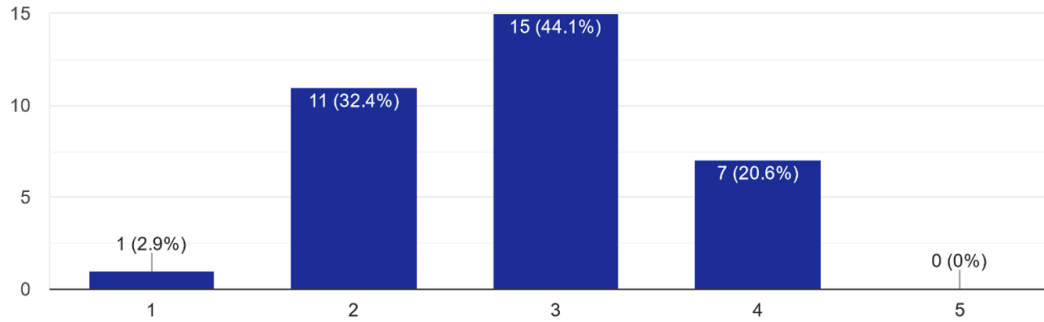


Figure 5.15.: Experience with Artificial Intelligence (AI) tools (Q2.5)

The following section discusses different criteria for rating the system. The participants rated the clarity of the summaries created by the AI with an average score of 4.00 ($SD=0.69$) out of 5 (see Figure 5.16, Q2.6). This high rating means that the summaries were generally clear and understandable for the users. The trust in the accuracy and reliability of the summaries was rated at an average of 3.41 ($SD=0.91$), as shown in Figure 5.17 (Q2.7). Although the result reflects a positive impression, it also makes it clear that trust can still be improved. With an average rating of 4.03 ($SD=0.89$), participants considered the system quite effective in tracing the summary content back to the original document (see Figure 5.18, Q2.8). This feature is precious for users who want to explore the original document's content in greater depth. Regarding coherence, the overall quality of the summaries received an average score of 4.09 ($SD=0.82$), as depicted in Figure 5.19 (Q2.9). This high rating indicates that the summaries maintained a logical progression of ideas, consistent tone, and clarity of connections. With an average score of 3.91 ($SD=0.82$) participants, the impact of the system's transparency on their general trust in the system was rated (see Figure 5.20, Q2.10). Improving transparency could further strengthen users' trust and make them more confident in the summaries created by the AI. The support provided by the explanation function when working with the summaries was rated 3.97 ($SD=0.75$) on average (see Figure 5.21, Q2.11). This indicates a high level of support and reflects the function's effectiveness in terms of the system's usefulness for users. Table 5.2 shows an overview of the participants' responses to the quantitative evaluation criteria.

5. Evaluation

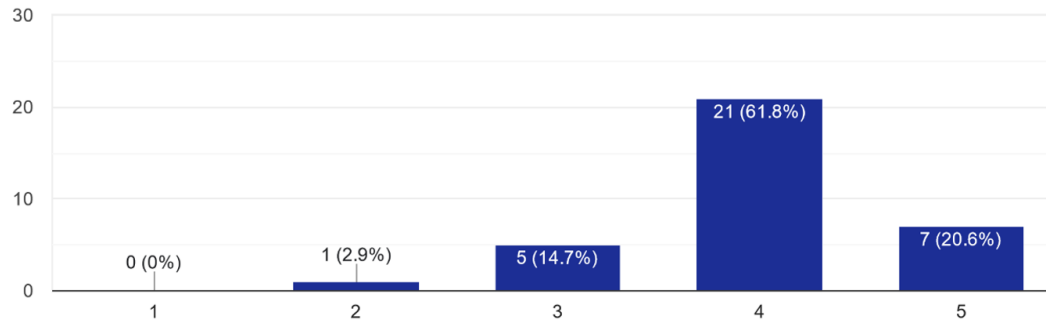


Figure 5.16.: Clarity of the AI-generated summaries (Q2.6)

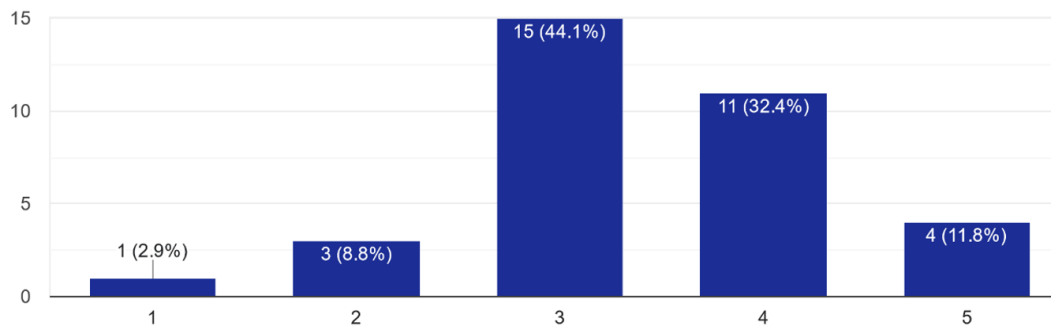


Figure 5.17.: To what extent the user trusts the AI-generated summaries to be accurate and reliable (Q2.7)

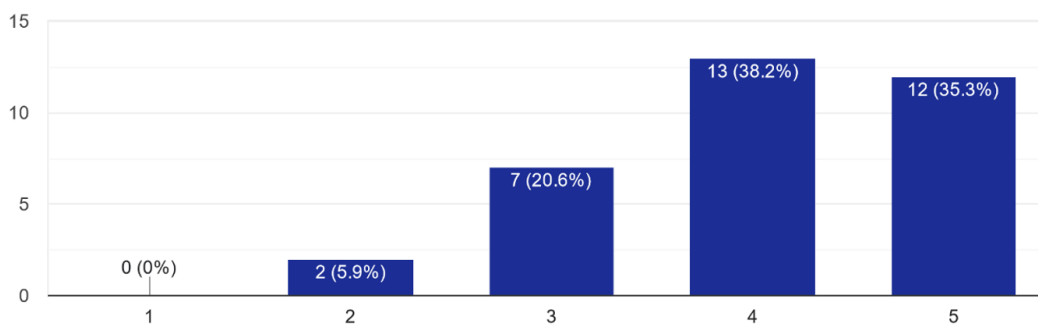


Figure 5.18.: Effectiveness of the system in allowing users to trace the summary content back to the original document (Q2.8)

5. Evaluation

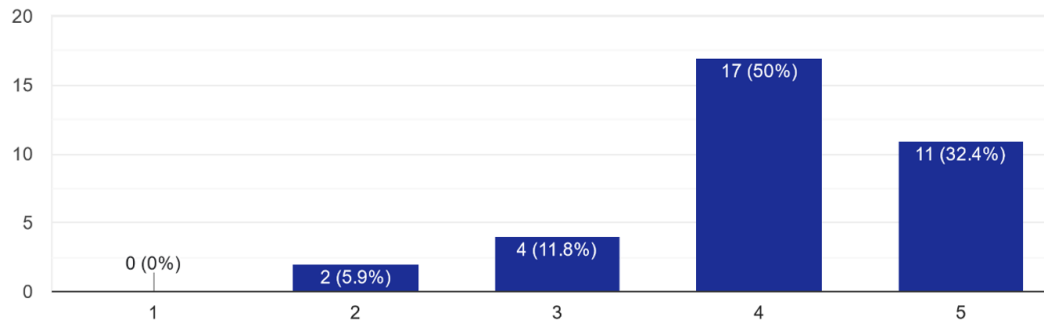


Figure 5.19.: Coherence of the AI-generated summary (Q2.9)

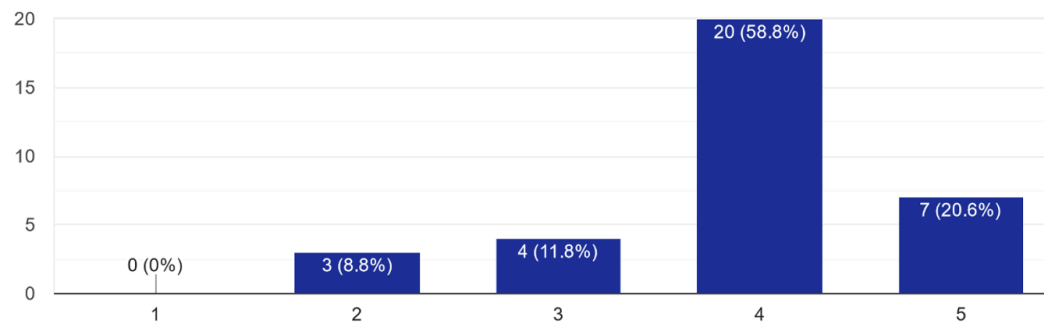


Figure 5.20.: The extent to which the transparency of the system contributes to general trust in the system (Q2.10)

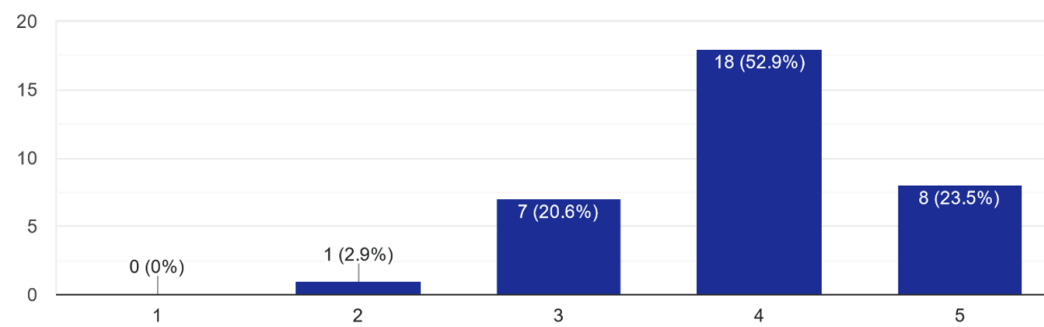


Figure 5.21.: The extent to which the AI system's explainability function supports work interactions with summaries (Q2.11)

5. Evaluation

Table 5.2.: Overview of the participants' responses to the quantitative evaluation criteria.

ID	Question	Mean	STD
Q2.6	How would you rate the clarity of the AI-generated summaries?	4.00	0.69
Q2.7	To what extent do you trust the summaries produced by the AI system to be accurate and reliable?	3.41	0.91
Q2.8	How effective do you think the system is in allowing users to trace the summary content back to the original document?	4.03	0.89
Q2.9	How would you rate the overall quality of the summaries in terms of their coherence? (Logical progression of ideas, Clarity of connections, Consistency of tone and style)	4.09	0.82
Q2.10	To what extent does the transparency of the system contribute to general trust in the system?	3.91	0.82
Q2.11	To what extent do you believe that the explainability function provided by the AI system supports you in your work when interacting with the summaries?	3.97	0.75

The answers to the open question (Q2.13) about the strengths and weaknesses of the summarization system revealed some interesting insights. Participants valued the system's ability to quickly summarize large documents, saving time and allowing information to be found more efficiently (*"It seems to extract the information precisely - saves time"; "Faster filtering and summarising of important information, which means you can work faster (time-saving)"*). The clarity, structured presentation, and easy comprehensibility of the summaries were highlighted as key strengths (*"Strengths are the clarity of the summarised sentences and the way the system ensures the user's trust by showing the most similar sentence, allowing the user to check the correctness."; "Clear, structured summary, easy to understand, sentences from the original version to compare for more transparency, time savings when working with texts"*). The function that allows users to trace the original text was also positively evaluated as it contributes to the transparency and trustworthiness of the system (*"The link to the original source makes the system more trustworthy."; "Generally improves trust in the summary"*). The efficiency of the system, which provides a quick overview of the data, and its flexibility of use were also mentioned as advantages (*"Efficient way to understand the idea of a paper and quick way to find the source of parts of the summarization, as the system works in the browser it can be easily deployed and integrated into various platforms"*).

However, some participants indicated that the system does not contain cohesive devices between sentences, resulting in a somewhat incoherent reading experience (*"Sometimes hard to read the sentences since they are not linked by cohesive devices in comparison to a paper"*). In addition, concerns were

expressed that the system could lead to false assumptions due to its lack of a critical perspective (*"No critical view, might come to "wrong" assumptions"*). Moreover, the simplicity of the writing style and the repetitive sentence structures were also identified as disadvantages (*"Monotonous, simple writing style, repeated sentence beginnings and similar sentence structure"*). The reliability of the summaries and the possibility of omitting essential points or topics due to the summarization process were also seen as weaknesses (*"The biggest weakness is that due to summarization probably important points/topics might be skipped"*; "Reliability").

The participants suggested several improvements to enhance the explainability and general functionality of the system (Q2.12). An overview of the most critical aspects is given below:

- Offering alternative summaries to provide different perspectives or more detailed information.
- Referencing specific pages or numbers in the summaries for better traceability.
- Highlighting the most similar sentence in the original document for better context understanding.
- Introducing an option for more extended summaries to build trust in the model by verifying the omitted content's relevance.
- Implementing a heat map or other visual aids to show which parts of the original paper were selected for the summary to provide more confidence in the accuracy of the summary.
- Providing the user with the option to show more than one sentence from the original document.

5.3. Limitations

With its explanation function, the AI summary system demonstrates a robust ability to create concise and transparent summaries from long scientific documents. However, the current version has some limitations. The system's dependence on Meta's large language model Llama2-7b-Chat model and the SBERT-based all-MiniLM-L6-v2 model can lead to incorrect output. The developed system is generally not faultless, and there is room for improvement. Several limitations can be derived from the results of the studies. The system only displays the currently semantically most similar sentence, and this sentence may contain too little information for the user. Furthermore, the summaries are not always coherent, and the user cannot

decide on the length of the summary or the number of sentences. Although the system allows users to trace the content back to the original document, the clarity of these connections varies. Moreover, confidence in the accuracy and reliability of the summaries was rated as moderately high but not optimal, suggesting that there is room for improvement in how the system interprets complex information. The demand for alternative summaries and other options shows that users want more customization from the AI summary system.

5.4. Summary

This chapter provides a detailed insight into evaluating the Summarization Quality Study and the Explainability Study. It sheds light on the effectiveness and transparency of the developed system, which uses the LLM Llama2-7b-Chat model to create summaries of scientific papers.

The Summarization Quality Study aims to evaluate AI-generated summaries' accuracy, coherence, and completeness. A total of 5 authors evaluated their papers. It revealed that authors of the original papers generally felt that the summaries accurately reflect the core content of their work, with high ratings in accuracy and the system's ability to highlight key research contributions. The overall satisfaction level of the study on the quality of the summaries was high, indicating a positive perception of the AI-generated content. However, the coherence was rated unevenly, indicating inconsistencies in the performance of the AI for different documents.

The Explainability Study focused on the system's transparency and impact on user trust. A total of 34 participants evaluated the system. It was shown that the system has succeeded mainly in making its summary process understandable to users, which has positively affected user confidence. The system was highlighted for its clarity and the quality of the summaries, and suggestions were made for further improvements to increase transparency and trust.

The participants in both studies had different demographic backgrounds and varying degrees of AI experience, which provided a broad spectrum of insights. When analyzing the results of both studies, it becomes clear that the system is able to produce informative summaries that reflect the original scientific papers, although there is still room for improvement in terms of coherence and transparency.

6. Lessons Learned

This chapter presents the insights gained from the literature review, the development phase, and the evaluation process.

6.1. Literature

The extensive literature review conducted for this thesis examined and researched several critical topics. The field of Explainable AI (XAI) and Natural Language Processing (NLP) is characterized by its fast pace, with new models and techniques emerging constantly. New models often arise while working on one topic that could change the research's conclusions or relevance. What was considered a state-of-the-art model for automatic text summarization can quickly become outdated and require constant adaptation.

Chapter 2, Background and Related Work, shows the progression from basic NLP techniques to the latest developments in automatic text summarization and XAI, as well as this field's rapid development and growing complexity. A recurring topic is the balance between improving models and maintaining or improving explainability. This balance is crucial for developing NLP systems that are powerful, transparent, and, therefore, trustworthy for the user. Examining different summarization techniques, evaluation metrics, and explainability methods shows the complexity of research in the field of NLP. This emphasizes the importance of explainable AI and automatic text summarization approaches for overcoming the challenges of future AI systems.

6.2. Development

Chapter 4, the development phase, was characterized by a continuous learning process. The interaction between the various system components, such as the Summarization Module and the Explainable AI Module, was crucial to achieving a functioning system.

Integrating complex models like Llama2-7B-Chat and SBERT-based model all-MiniLM-L6-v2 into the system architecture presented challenges. The maximum input sequence of 4096 tokens revealed that the basic model was not sufficient to summarize long documents. These experiences make it clear how important it is to know precisely the models' possibilities and limitations. The project successfully used the Large Language Model Llama2-7B-Chat with an NVIDIA A100 Tensor Core GPU, which ensures fast processing and high-quality results. Meta's Llama2-7B-Chat model and Gradio for the user interface emphasize integrating cutting-edge technology to improve system performance and usability.

The project faced the challenge of managing large amounts of text, which required a chunking mechanism for text processing and the use of `RecursiveCharacterTextSplitter`. The ability of AI summarization systems to efficiently process large amounts of data and their modularity for easy updates are notable accomplishments. The project's development process also highlights the fast-moving nature of the NLP field, where constant adaptation to the latest developments can improve performance and capabilities.

6.3. Evaluation

Chapter 5, the evaluation phase, consisting of the Summarization Quality Study and the Explainability Study, provided critical insights into the performance of the AI summary system. The evaluation studies demonstrate the system's overall effectiveness in summarizing complex scientific documents and providing explainable outputs that build user trust. The AI system using the Llama2-7b-Chat model was generally effective in producing summaries that accurately reflected the critical points of the original scientific papers. Most participants felt that the length of the summaries was appropriate, indicating that the system is able to summarize information effectively.

In general, it was very important to explain the system to users as clearly as possible with pictures and a video so that even users with no technical background could immediately understand the functionality. Furthermore, the questionnaires were revised in many iterations in order to prevent ambiguity and to be able to answer the research questions. The feedback from the open questions revealed areas where improvements could be made, such as enhancing coherence and transparency. The participants made valuable suggestions for improving the system, for example, alternative summaries, references to specific parts of the original document, and the integration of visual aids such as heat maps. The demographic characteristics of the participants and the different satisfaction levels indicate that the user experience can be subjective. Therefore, future studies should consider even more participants.

7. Conclusion and Future Work

This chapter summarizes the insights gained from this work and looks at possible further improvements to the AI summary system and perspectives for future research.

7.1. Conclusion

This research focused on developing an AI summarization system leveraging Meta's Llama2-7B-Chat with an NVIDIA A100 Tensor Core GPU. The system's architecture, optimized for processing long scientific papers, ensures efficient text summarization. The system effectively enhances user confidence by integrating an explanation mechanism and utilizing embeddings from the SBERT-based model all-MiniLM-L6-v2.

The evaluation included the Summarization Quality Study and the Explainability Study. The former assessed the quality of the summaries produced by the AI in terms of accuracy, coherence, and completeness. The Summarization Quality Study revealed a high level of satisfaction among the authors, with the system's accuracy in reproducing the original papers' content scoring an average of 4.20 (STD=0.40) out of 5. The capability to highlight essential research contributions was also rated positively, with an average score of 4.00 (STD= 1.26). However, with an average score of 3.40 (STD=1.02), the coherence of the summaries showed more variation, suggesting that there is room for improvement in ensuring consistent quality across all summaries.

The Explainability Study, which had 34 participants, focused on the system's transparency and impact on user trust. The results showed that the system made the summary process more understandable to users, positively affecting their confidence. The clarity of the AI-generated summaries scored an average of 4.00 (STD=0.69). The system's effectiveness in allowing users to trace summary content back to the original document was also rated well,

with an average score of 4.03 (STD=0.89). Nonetheless, trust in the accuracy and reliability of the summaries indicated growth potential, with a score of 3.41 (STD=0.91).

However, the system's reliance on Meta's large language model Llama2-7b-Chat and the SBERT-based all-MiniLM-L6-v2 model can result in incorrect output. The system currently only displays the semantically most similar sentence, and this sentence may contain too little information or may not be precise enough for the user. The developed AI summary system with an explanation function is not faultless, and there is room for improvement.

7.2. Future Work

This work provides several opportunities for future research and improvement. Updating the system to incorporate the latest models and techniques will be critical to maintaining its effectiveness and relevance. In addition, it is crucial to consider the variability of coherence assessments. Offering alternative summaries could provide users with different perspectives or more detailed information on specific sections of the original document.

Another potential area for future work is to expand the system's explanatory capabilities to increase its trust score. While the current system provides a foundational level of explainability, other methods, such as advanced interactive visualization techniques, referencing specific pages, or highlighting the most similar sentence in the original document, could further increase user trust.

In addition, it would also be ideal if the system were no longer dependent on Meta's large language model Llama2-7b-Chat, as this would influence the generated summaries more, subsequently increasing transparency. It would require many resources to build a model from scratch. Therefore, another possibility would be to update the system to the latest open source models, which could easily be possible due to the modularity, and to optimize the system from this basis.

Furthermore, the ability to customize the length of the summaries or select more than one similar sentence could increase user satisfaction and the system's usefulness. In addition, integrating user feedback mechanisms could provide invaluable data for continuous improvement.

7. Conclusion and Future Work

The evaluation studies also emphasized the need for broader and more diverse test scenarios. Future work could include more extensive studies with a broader range of participants from different professional backgrounds to better understand the system's performance and user experience. The foundation laid by this work provides a solid basis for future developments.

Bibliography

- Abujabal, A., Saha Roy, R., Yahya, M., & Weikum, G. (2017). QUINT: Interpretable question answering over knowledge bases. In L. Specia, M. Post, & M. Paul (Eds.), *Proceedings of the 2017 conference on empirical methods in natural language processing: System demonstrations* (pp. 61–66). Association for Computational Linguistics. <https://doi.org/10.18653/v1/D17-2011>. (Cit. on pp. 35, 37)
- Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6, 52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052> (cit. on pp. 34, 35)
- Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., & Hajishirzi, H. (2019). MathQA: Towards interpretable math word problem solving with operation-based formalisms. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 2357–2367). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1245>. (Cit. on p. 35)
- Arya, V., Bellamy, R. K., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S. C., Houde, S., Liao, Q. V., Luss, R., Mojsilović, A., et al. (2019). One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques. *arXiv preprint arXiv:1909.03012* (cit. on pp. xxii, 33, 34).
- Awasthi, I., Gupta, K., Bhogal, P. S., Anand, S. S., & Soni, P. K. (2021). Natural language processing (nlp) based text summarization - a survey, 1310–1317. <https://doi.org/10.1109/ICICT50816.2021.9358703> (cit. on pp. xxii, 6, 7, 11–13)
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (cit. on pp. xix, 11, 21, 34, 36).
- Balakrishnan, V., & Lloyd-Yemoh, E. (2014). Stemming and lemmatization: A comparison of retrieval performances (cit. on p. 9).
- Bandy, J., & Vincent, N. (2021). Addressing “documentation debt” in machine learning: A retrospective datasheet for bookcorpus. *Thirty-fifth*

- Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)* (cit. on p. 25).
- Banerjee, S., & Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72 (cit. on pp. 30, 31).
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3, 993–1022 (cit. on p. 17).
- Brants, T. (2000). Tnt-a statistical part-of-speech tagger. *arXiv preprint cs/0003055* (cit. on p. 10).
- Brill, E. (1992). A simple rule-based part of speech tagger. *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992* (cit. on p. 10).
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Nee-lakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901 (cit. on pp. 7, 23).
- Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., et al. (2005). The ami meeting corpus: A pre-announcement. *International workshop on machine learning for multimodal interaction*, 28–39 (cit. on pp. 24, 25).
- Carton, S., Mei, Q., & Resnick, P. (2018). Extractive adversarial networks: High-recall explanations for identifying personal attacks in social media posts. *arXiv preprint arXiv:1809.01499* (cit. on p. 38).
- Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8), 832 (cit. on pp. xxii, 38).
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (cit. on pp. 20, 21).
- Chomsky, N. (1957). *Syntactic structures*. De Gruyter Mouton. <https://doi.org/doi:10.1515/9783112316009>. (Cit. on p. 5)
- Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 93–98 (cit. on p. 21).
- Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., & Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685* (cit. on pp. 24, 26).
- Croce, D., Rossini, D., & Basili, R. (2019). Auditing deep learning processes through kernel-based explanatory models. *Proceedings of the 2019*

- Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4037–4046 (cit. on pp. 35, 37).
- Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., & Sen, P. (2020). A survey of the state of explainable ai for natural language processing. *AACL-IJCNLP 2020* (cit. on pp. 36–39).
- Deerwester, S., Dumais, S., Furnas, G., Landauer, T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 391–407 (cit. on p. 17).
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (cit. on pp. 7, 22, 41, 58).
- Dong, Y., Li, Z., Rezagholizadeh, M., & Cheung, J. C. K. (2019). Editnts: An neural programmer-interpreter model for sentence simplification through explicit editing. *arXiv preprint arXiv:1906.08104* (cit. on p. 39).
- Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (cit. on p. 33).
- Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4), 325–327 (cit. on p. 35).
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179–211 (cit. on p. 6).
- Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22, 457–479 (cit. on pp. 17, 18).
- Fang, Y., & Teufel, S. (2016). Improving argument overlap for proposition-based summarisation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 479–485 (cit. on p. 25).
- Farmakiotou, D., Karkaletsis, V., Koutsias, J., Sigletos, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000). Rule-based named entity recognition for greek financial texts. *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, 75–78 (cit. on p. 10).
- Fattah, M. A., & Ren, F. (2009). Ga, mr, ffn, pnn and gmm based models for automatic text summarization. *Computer Speech Language*, 23(1), 126–144. <https://doi.org/10.1016/j.csl.2008.04.002> (cit. on pp. xxii, 13, 14)
- Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review*, 47. <https://doi.org/10.1007/s10462-016-9475-9> (cit. on p. 13)

- Gatt, A., & Reiter, E. (2009). Simplenlg: A realisation engine for practical applications. *Proceedings of the 12th European workshop on natural language generation (ENLG 2009)*, 90–93 (cit. on p. 19).
- Genest, P.-E., & Lapalme, G. (2011). Framework for abstractive summarization using text-to-text generation. *Proceedings of the workshop on monolingual text-to-text generation*, 64–73 (cit. on p. 19).
- Genest, P.-E., & Lapalme, G. (2012). Fully abstractive approach to guided summarization. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 354–358. <https://aclanthology.org/P12-2069> (cit. on pp. 18, 19)
- Godin, F., Demuynck, K., Dambre, J., De Neve, W., & Demeester, T. (2018). Explaining character-aware neural networks for word-level prediction: Do they discover linguistic rules? *arXiv preprint arXiv:1808.09551* (cit. on p. 34).
- Goodman, B., & Flaxman, S. (2017). European union regulations on algorithmic decision-making and a “right to explanation”. *AI magazine*, 38(3), 50–57 (cit. on p. 33).
- Graff, D., Kong, J., Chen, K., & Maeda, K. (2003). English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1), 34 (cit. on pp. 24, 26).
- Graham, Y. (2015). Re-evaluating automatic summarization with bleu and 192 shades of rouge. *Proceedings of the 2015 conference on empirical methods in natural language processing*, 128–137 (cit. on p. 29).
- Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5), 1–42 (cit. on pp. 31, 33).
- Harabagiu, S. M., & Lacatusu, F. (2002). Generating single and multi-document summaries with gistexter. *Document Understanding Conferences*, 11–12 (cit. on p. 19).
- Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28 (cit. on pp. 24, 26).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780 (cit. on pp. 6, 20).
- Hovy, E., & Lin, C.-Y. (1998). Automated text summarization and the summarist system. *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, 197–214. <https://doi.org/10.3115/1119089.1119121> (cit. on pp. xxii, 12, 13)
- Hu, B., Chen, Q., & Zhu, F. (2015). Lcsts: A large scale chinese short text summarization dataset. *arXiv preprint arXiv:1506.05865* (cit. on p. 25).
- Hutchins, W. J. (2004). The georgetown-ibm experiment demonstrated in january 1954. *Conference of the Association for Machine Translation in the Americas*, 102–114 (cit. on p. 5).

- Jelinek, F. (1985). The development of an experimental discrete dictation recognizer. *Proceedings of the IEEE*, 73(11), 1616–1624 (cit. on p. 5).
- Ježek, K., & Steinberger, J. (2008). Automatic text summarization (the state of the art 2007 and new challenges). *Proceedings of Znalosti*, 1–12 (cit. on p. 11).
- Jones, K. S. (1998). Automatic summarising: Factors and directions. *ArXiv, cmp-lg/9805011* (cit. on pp. xxii, 12, 13).
- Jones, K. S. (2007). Automatic summarising: The state of the art. *Information Processing & Management*, 43(6), 1449–1481 (cit. on p. 27).
- Jones, K. S., & Galliers, J. R. (1995). Evaluating natural language processing systems: An analysis and review (cit. on p. 27).
- Jurafsky, D., & Martin, J. (2023). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (Vol. 3). (Cit. on pp. 8–10).
- Kannan, S., Gurusamy, V., Vijayarani, S., Ilamathi, J., Nithya, M., Kannan, S., & Gurusamy, V. (2014). Preprocessing techniques for text mining. *International Journal of Computer Science & Communication Networks*, 5(1), 7–16 (cit. on pp. 7, 8).
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (cit. on p. 6).
- Kiss, T., & Strunk, J. (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4), 485–525. <https://doi.org/10.1162/coli.2006.32.4.485> (cit. on p. 8)
- Kryściński, W., Rajani, N., Agarwal, D., Xiong, C., & Radev, D. (2021). Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209* (cit. on p. 25).
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (cit. on p. 23).
- Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1), 50–70 (cit. on p. 10).
- Li, J., Chen, X., Hovy, E., & Jurafsky, D. (2015). Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066* (cit. on p. 34).
- Liddy, E. D. (2001). Natural language processing. In *Encyclopedia of library and information science* (2nd ed., pp. 2126–2127). Marcel Dekker, Inc. (Cit. on p. 4).
- Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text summarization branches out*, 74–81 (cit. on pp. xxii, 29, 30).

- Lin, H., & Ng, V. (2019). Abstractive summarization: A survey of the state of the art. *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 9815–9822 (cit. on p. 18).
- Lloret, E., Plaza, L., & Aker, A. (2018). The challenging task of summary evaluation: An overview. *Language Resources and Evaluation*, 52, 101–148 (cit. on pp. 27, 28).
- Lopez, A. (2008). Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3), 1–49 (cit. on p. 11).
- Lovins, J. B. (1968). Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, 11(1-2), 22–31 (cit. on p. 9).
- Ma, K., Tian, M., Tan, Y., Xie, X., & Qiu, Q. (2022). What is this article about? generative summarization with the bert model in the geosciences domain. *Earth Science Informatics*, 15, 1–16. <https://doi.org/10.1007/s12145-021-00695-2> (cit. on pp. 14, 15)
- MacCarthy, M. (2019). An examination of the algorithmic accountability act of 2019. *Available at SSRN 3615731* (cit. on p. 33).
- Malan, R., Bredemeyer, D., et al. (2001). Functional requirements and use cases. *Bredemeyer Consulting* (cit. on pp. 44, 46, 47).
- Malyusz, L., Hajdu, M., & Vattai, Z. (2021). Comparison of different algorithms for time analysis for cpm schedule networks. *Automation in Construction*, 127, 103697. [https://doi.org/https://doi.org/10.1016/j.autcon.2021.103697](https://doi.org/10.1016/j.autcon.2021.103697) (cit. on p. 12)
- Mann, W. C., & Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*, 8, 243–281. <https://api.semanticscholar.org/CorpusID:60514661> (cit. on pp. xxii, 15, 16)
- Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press. (Cit. on p. 5).
- Mehdad, Y., Carenini, G., & Ng, R. (2014). Abstractive summarization of spoken and written conversations based on phrasal queries. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1220–1230 (cit. on p. 18).
- Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. *Proceedings of the 2004 conference on empirical methods in natural language processing*, 404–411 (cit. on pp. 17, 18).
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (cit. on p. 6).
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM computing surveys (CSUR)*, 54(3), 1–40 (cit. on p. 11).

- Moawad, I., & Aref, M. (2012). Semantic graph reduction approach for abstractive text summarization. *Proceedings - ICCES 2012: 2012 International Conference on Computer Engineering and Systems*, 132–138. <https://doi.org/10.1109/ICCES.2012.6408498> (cit. on p. 19)
- Morwal, S., Jahan, N., & Chopra, D. (2012). Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol, 1* (cit. on p. 10).
- Mullenbach, J., Wiegrefe, S., Duke, J., Sun, J., & Eisenstein, J. (2018). Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695* (cit. on p. 34).
- Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., et al. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023* (cit. on pp. 23, 24).
- Nenkova, A., & McKeown, K. (2011). *Automatic summarization* (Vol. 5). <https://doi.org/10.1561/15000000015>. (Cit. on pp. xxii, 12, 13)
- Nenkova, A., Passonneau, R., & McKeown, K. (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(2), 4–es (cit. on pp. 27, 28).
- NIST. (2023). Text analysis conference [Last updated: May 5, 2023]. *Text Analysis Conference (TAC)*. <https://tac.nist.gov>. (Cit. on pp. 24, 26)
- Norkute, M., Herger, N., Michalak, L., Mulder, A., & Gao, S. (2021). Towards explainable ai: Assessing the usefulness and impact of added explainability features in legal document summarization. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–7 (cit. on p. 41).
- OpenAI. (2023). Gpt-4 technical report. *ArXiv, abs/2303.08774*. <https://api.semanticscholar.org/CorpusID:257532815> (cit. on pp. 7, 23)
- Over, P., Dang, H., & Harman, D. (2007). Duc in context. *Information Processing & Management*, 43(6), 1506–1520 (cit. on pp. 23, 26).
- Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. *Proceedings of the 7th International World Wide Web Conference*, 161–172. citeseer.nj.nec.com/page98pagerank.html (cit. on p. 17)
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318 (cit. on p. 28).
- Parida, S., & Motlicek, P. (2019). Abstract text summarization: A low resource challenge. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 5994–5998 (cit. on p. 31).

- Pedreschi, D., Giannotti, F., Guidotti, R., Monreale, A., Ruggieri, S., & Turini, F. (2019). Meaningful explanations of black box ai decision systems. *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 9780–9784 (cit. on p. 33).
- Perez-Ortiz, J. A., & Forcada, M. L. (2001). Part-of-speech tagging with recurrent neural networks. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, 3, 1588–1592 (cit. on p. 10).
- Pezeshkpour, P., Tian, Y., & Singh, S. (2019). Investigating robustness and interpretability of link prediction via adversarial modifications. *arXiv preprint arXiv:1905.00563* (cit. on p. 37).
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137 (cit. on p. 9).
- Quinlan, J. R. (1990). Decision trees and decision-making. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 339–346 (cit. on pp. xix, 32).
- Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training (cit. on pp. 7, 22).
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9 (cit. on pp. 7, 23).
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1), 5485–5551 (cit. on p. 23).
- Rajani, N. F., McCann, B., Xiong, C., & Socher, R. (2019). Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361* (cit. on pp. 37, 38).
- Ramasubramanian, C., & Ramya, R. S. (2013). Effective pre-processing activities in text mining using improved porter's stemming algorithm. <https://api.semanticscholar.org/CorpusID:55095471> (cit. on p. 9)
- Rane, N., & Govilkar, S. (2019). Recent trends in deep learning based abstractive text summarization. *Int. J. Recent Technol. Eng*, 8(3), 3108–3115 (cit. on p. 18).
- Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (cit. on p. 42).
- Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1), 57–87 (cit. on p. 37).
- Ribeiro, A. H., Tiels, K., Aguirre, L. A., & Schön, T. (2020). Beyond exploding and vanishing gradients: Analysing rnn training using attractors and smoothness. *International conference on artificial intelligence and statistics*, 2370–2380 (cit. on p. 20).

- Ribeiro, M., Singh, S., & Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In J. DeNero, M. Finlayson, & S. Reddy (Eds.), *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Demonstrations* (pp. 97–101). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N16-3020>. (Cit. on pp. 34, 35)
- Ross, A. S., Hughes, M. C., & Doshi-Velez, F. (2017). Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717* (cit. on p. 39).
- Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (cit. on pp. 21, 24, 40).
- Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078* (cit. on p. 6).
- Schwarzenberg, R., Harbecke, D., Macketanz, V., Avramidis, E., & Möller, S. (2019). Train, sort, explain: Learning to diagnose translation models. In W. Ammar, A. Louis, & N. Mostafazadeh (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics (demonstrations)* (pp. 29–34). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-4006>. (Cit. on p. 36)
- Scott, A. C., Clancey, W. J., Davis, R., & Shortliffe, E. H. (1977). Explanation capabilities of production-based consultation systems. *American Journal of Computational Linguistics*, 1–50 (cit. on p. 32).
- Shiwen, Y., & Xiaojing, B. (2014). Rule-based machine translation. In *Routledge encyclopedia of translation technology* (pp. 186–200). Routledge. (Cit. on p. 11).
- Sisodia, Y. (2022). Explainable ai for nlp: Decoding black box. *International Journal of Computer Trends and Technology*, 70(7), 11–15. <https://doi.org/10.14445/22312803/IJCTT-V70I7P103> (cit. on p. 35)
- Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications*, 78, 857–875 (cit. on pp. 20, 21).
- Suleiman, D., & Awajan, A. (2020). Deep learning based abstractive text summarization: Approaches, datasets, evaluation measures, and challenges. *Mathematical problems in engineering*, 2020, 1–29 (cit. on p. 20).
- Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27 (cit. on p. 6).

- Sydorova, A., Poerner, N., & Roth, B. (2019). Interpretable question answering on knowledge bases and text. *arXiv preprint arXiv:1906.10924* (cit. on p. 39).
- Tabassum, A., & Patil, R. R. (2020). A survey on text pre-processing & feature extraction techniques in natural language processing. *International Research Journal of Engineering and Technology (IRJET)*, 7(06), 4864–4867 (cit. on p. 9).
- Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (cit. on pp. 23, 42).
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30 (cit. on pp. xix, 11, 21, 22, 40, 41).
- Vig, J. (2019). A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714* (cit. on p. 41).
- Vijayan, V. K., Bindu, K., & Parameswaran, L. (2017). A comprehensive study of text classification algorithms. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1109–1113 (cit. on p. 10).
- Wang, H., Gao, Y., Bai, Y., Lapata, M., & Huang, H. (2021). Exploring explainable selection to control abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15), 13933–13941 (cit. on p. 40).
- Wang, S., Zhao, X., Li, B., Ge, B., & Tang, D. (2017). Integrating extractive and abstractive models for long text summarization. *2017 IEEE international congress on big data (BigData congress)*, 305–312 (cit. on p. 29).
- Weizenbaum, J. (1966). Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1), 36–45 (cit. on p. 5).
- Xiao, W., & Carenini, G. (2019). Extractive summarization of long documents by combining global and local context. *arXiv preprint arXiv:1909.08089* (cit. on p. 30).
- Xie, Q., Ma, X., Dai, Z., & Hovy, E. (2017). An interpretable knowledge transfer model for knowledge base completion. *arXiv preprint arXiv:1704.05908* (cit. on pp. 34, 39).
- Zhang, J., Zhao, Y., Saleh, M., & Liu, P. (2020). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. *International Conference on Machine Learning*, 11328–11339 (cit. on pp. 18, 23).
- Zhu, L., Wang, W., Huang, M., Chen, M., Wang, Y., & Cai, Z. (2022). A n-gram based approach to auto-extracting topics from research articles¹.

- Journal of Intelligent & Fuzzy Systems*, 43(5), 6137–6146 (cit. on pp. 28, 29).
- Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *Proceedings of the IEEE international conference on computer vision*, 19–27 (cit. on p. 25).

Appendix

Appendix A.

Screenshots of the system's output

A.1. Generated summaries and explainability

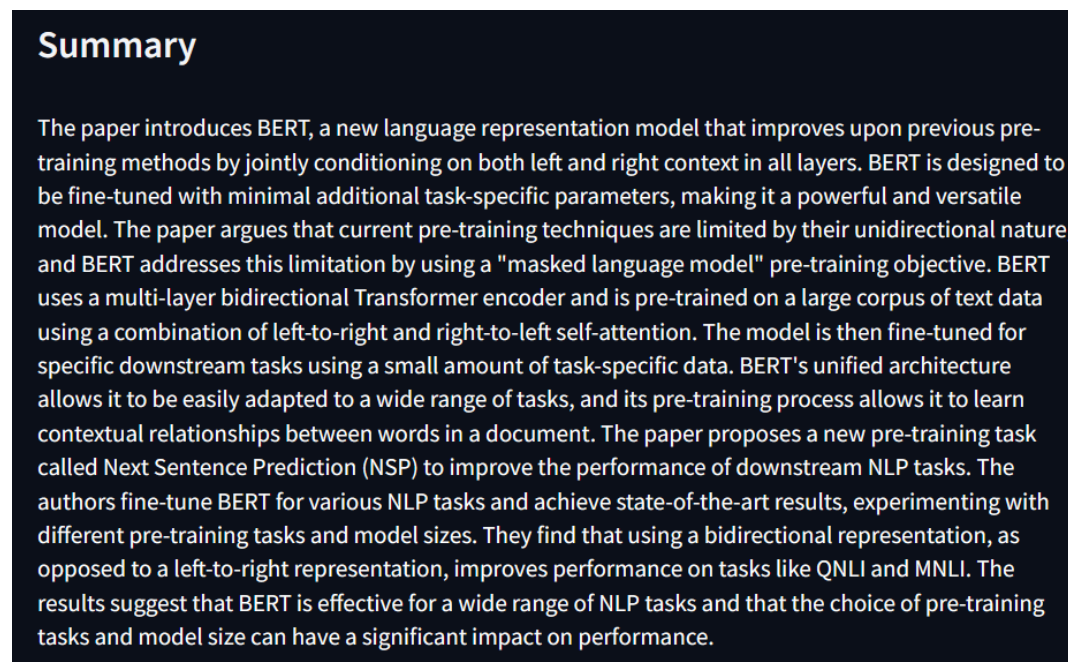


Figure A.1.: Summary 1

Appendix A. Screenshots of the system's output

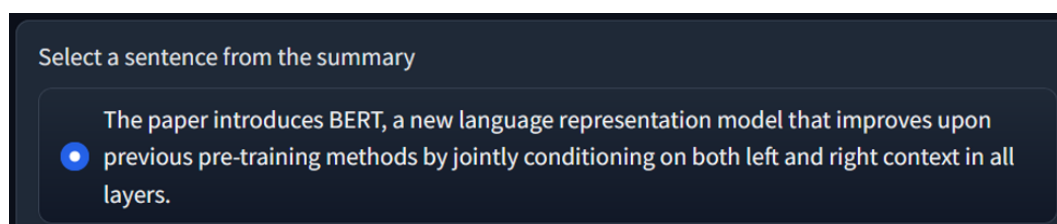


Figure A.2.: Selected Summary 1 Sentence

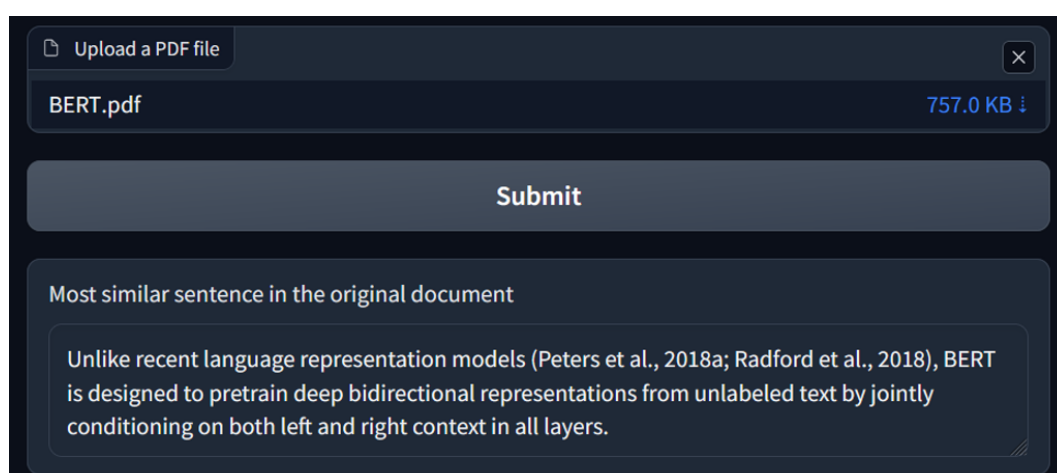


Figure A.3.: Most similar sentence in the original document 1

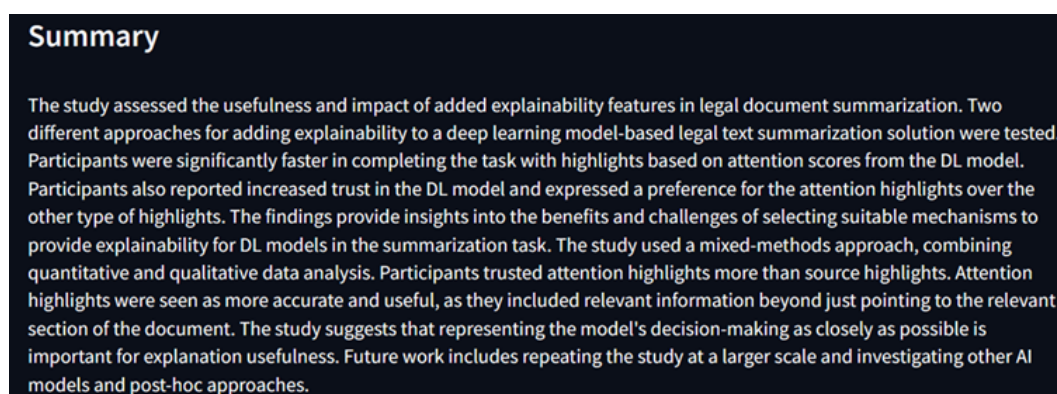


Figure A.4.: Summary 2

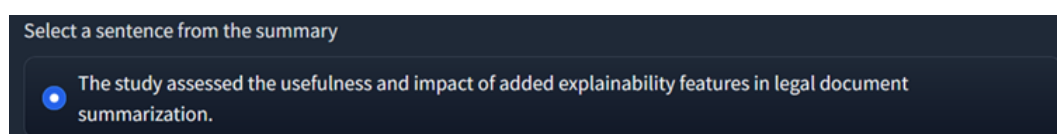


Figure A.5.: Selected Summary 2 Sentence

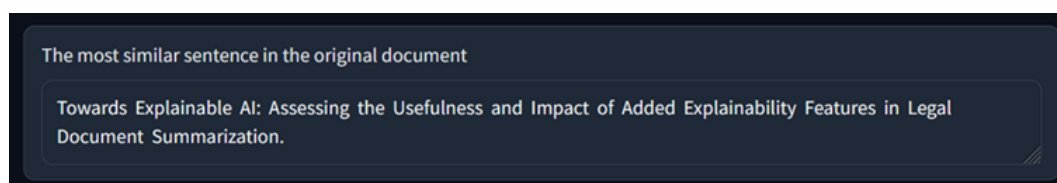


Figure A.6.: Most similar sentence A in the original document 2

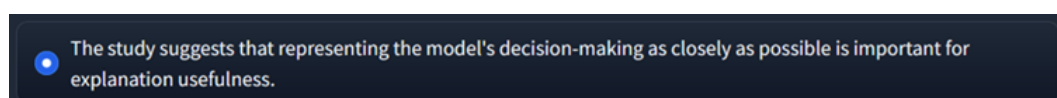


Figure A.7.: Another selected Summary 2 Sentence

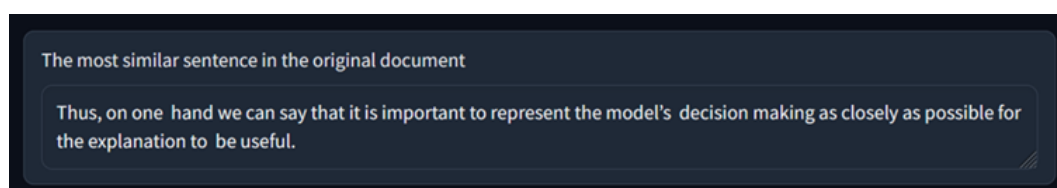


Figure A.8.: Most similar sentence B in the original document 2

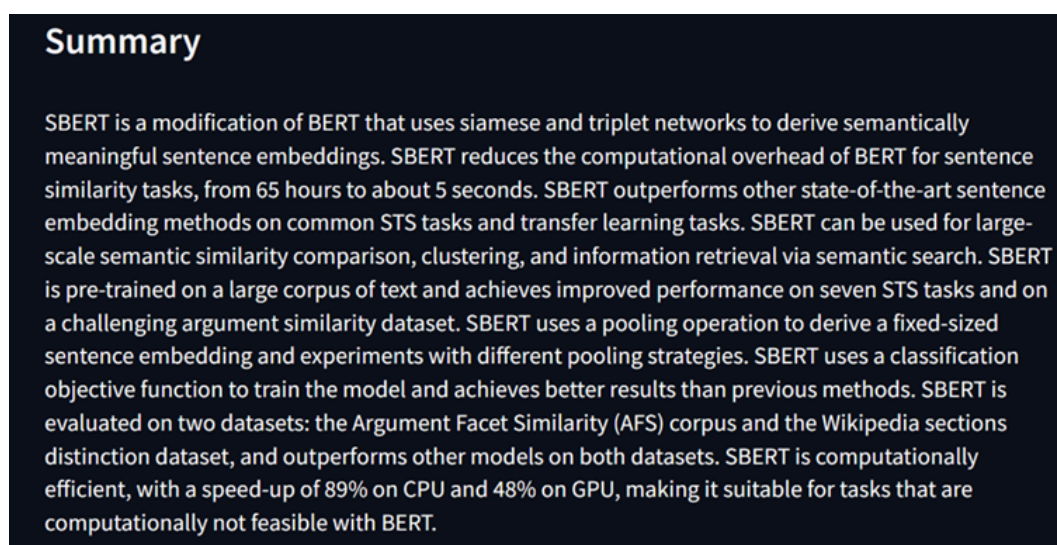


Figure A.9.: Summary 3

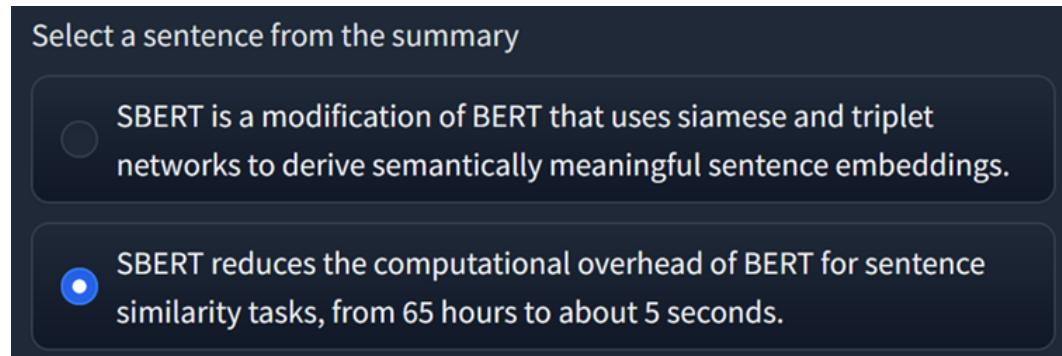


Figure A.10.: Selected Summary 3 Sentence

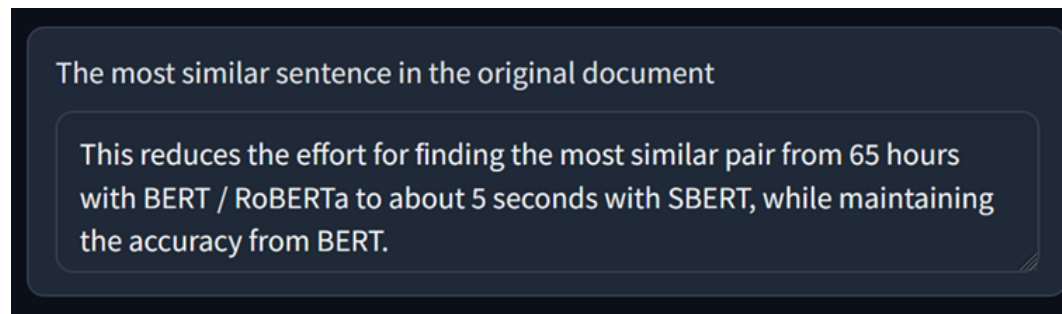


Figure A.11.: Most similar sentence A in the original document 3

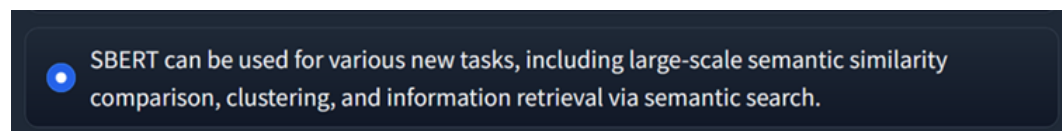


Figure A.12.: Another selected Summary 3 Sentence

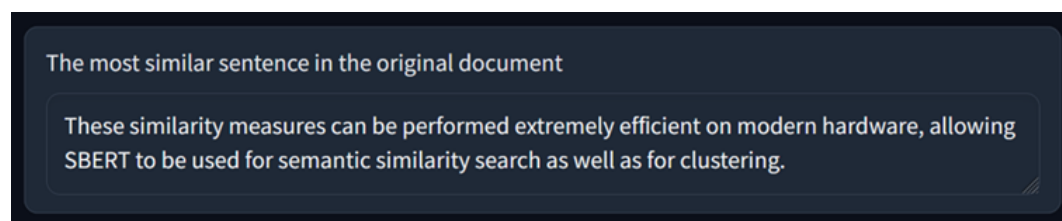


Figure A.13.: Most similar sentence B in the original document 3