Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Helene Janine Stang
Ingeborg Sætersdal Sollid

A Hybrid Multi-document Summarization System for Biomedical Articles

Master's thesis in Computer Science
Supervisor: Heri Ramampiaro
May 2021

NTNU
Norwegian University of
Science and Technology

Helene Janine Stang
Ingeborg Sætersdal Sollid

A Hybrid Multi-document Summarization System for Biomedical Articles

Master's thesis in Computer Science
Supervisor: Heri Ramampiaro
May 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

The main objective of this work is to investigate how text summarization can be used to support decision-making in the biomedical domain, especially in the diagnosis of cerebral palsy. Machine learning has shown great potential for the early diagnosis of CP. For the medical experts to better understand the system's predictions, articles related to the algorithm's findings will be retrieved. Automatic summarization of these articles can help medical experts save valuable time and provide essential information to support the decision of the final diagnosis. In recent years, natural language processing has seen significant advances in the use of neural-network-based methods. The availability of pre-trained language models has resulted in a significant improvement in automatic text summarization. However, it remains challenging to create text summaries of multiple long documents in the biomedical domain close to how humans would have written them.

We propose a novel biomedical multi-document summarization system consisting of an extractive-abstractive summarizer. The extractive step utilizes various text mining techniques, while the abstractive step employs a pre-trained language model. Our main focus is the extractive part, as it enables the summarization of multiple documents by reducing the input text of the pre-trained model. The system should handle redundant, complementary, and conflicting information within the biomedical domain and produce concise and consistent summaries. In order to find the optimal summarization pipeline, we conduct an ablation study. This study involves experiments with different techniques within representation, clustering, scoring, and selection of sentences. The evaluation of our proposed approach system shows great potential for supporting decision-making within the biomedical domain and validating predictions from machine learning models. The generated summaries look generally good, although they still suffer from some redundancy and conflicting information, so the remaining challenges need to be solved in future work.

Sammendrag

Hovedmålet med dette arbeidet er å undersøke hvordan tekstsammendrag kan brukes til å støtte beslutningsprosesser i det biomedisinske domenet, spesielt for diagnostisering cerebral parese. Maskinlæring har vist et stort potensiale for tidlig diagnostisering av CP. For at medisinske eksperter skal forstå systemets prediksjoner bedre vil artikler relatert til algoritmens funn bli hentet ut. Automatisk oppsummering av disse artiklene kan hjelpe medisinske eksperter med å spare verdifull tid og gi viktig informasjon for å støtte beslutningen av den endelige diagnosen. De siste årene har naturlig språkprosessering (NLP) sett betydelige fremskritt i bruken av nevrale nettverksbaserte metoder. Tilgjengeligheten av forhåndstrente språkmodeller har resultert i en betydelig forbedring i automatisk tekstoppsummering. Det er imidlertid fortsatt utfordrende å lage tekstsammendrag av flere lange dokumenter innen det biomedisinske domenet som er nær hvordan mennesker ville ha skrevet dem.

Vi presenterer et nytt system for oppsummering av flere biomedisinske dokumenter som består av en ekstraktiv-abstraktiv oppsummerer. Det ekstraktive steget benytter forskjellige teknikker innen text mining, mens det abstraktive trinnet benytter en forhåndstrent språkmodell. Vårt hovedfokus er den ekstraktive delen, da den muliggjør oppsummering av flere dokumenter ved å redusere mengden tekst som sendes inn til den forhåndstrente modellen. Systemet skal håndtere overflødig og motstridende informasjon innenfor det biomedisinske domenet og produsere konsise og konsistente sammendrag. For å finne det optimale oppsummeringssystemet gjennomfører vi et ablasjonsstudie. Dette studiet involverer eksperimenter med ulike teknikker innen representasjon, gruppering, scoring og utvelging av setninger. Evalueringen av det foreslåtte systemet vårt viser et stort potensiale for å støtte beslutningsprosesser innen det biomedisinske domenet og validere prediksjoner fra maskinlæringsmodeller. Oppsummeringene som genereres ser generelt bra ut, men lider imidlertid fortsatt av overflødig og motstridende informasjon, så disse gjenværende utfordringene må løses i fremtidig arbeid.

Preface and Acknowledgement

This master thesis is written in collaboration between Helene Janine Stang and Ingeborg Sætersdal Sollid to complete a five-year Master of Science degree in Computer Science at the Norwegian University of Science and Technology. The thesis is based on a specialization project that was carried out throughout the autumn of 2020, and therefore includes some of its relevant parts.

The research conducted is part of a larger research project in collaboration between the Norwegian University of Science and Technology and St. Olavs University Hospital. We would like to thank our supervisor Professor Heri Ramampiaro for valuable discussions and feedback. We are sincerely grateful for your guidance and for keeping us motivated throughout this project. We would also like to thank Researcher Lars Adde for his contribution to our master thesis.

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

Cerebral palsy is the most common movement disorder for children. Traditionally, CP diagnosis has been made at the age of two years, but detecting it at an earlier stage can improve cognitive and motoric functions (Adde, 2019). The existing solution for predicting CP is limited by the need for expensive equipment and highly experienced personnel (Adde et al., 2010). In a collaboration between St. Olav's University Hospital and the Norwegian University of Science and Technology, the In-Motion project aims to develop machine learning techniques to predict CP in infants. The system's prediction can support the medical expert's decisions in diagnosing an infant, but medical experts must verify and understand the prediction. To trust the predictions blindly would be irresponsible. A wrong decision can be very harmful and affect human life. Therefore it is important to explain why the system decided on the prediction. Relevant keywords describing the prediction would be optimal output from the machine learning algorithm. One of the attempts to further explain the prediction would be to retrieve articles based on the keywords. Natural language processing techniques like automatic summarization hold promise for extracting decision-support information from text (Workman et al., 2012). Therefore, a summary to structure and compress the multiple articles retrieved is desired. Automatic summarization can help medical experts reduce valuable time and hopefully provide essential information to support the final diagnosis decision. Our idea to fulfill this is a hybrid summarization system that utilizes NLP and text mining techniques to summarize biomedical articles.

In addition to the In-Motion system, automatic summarization of biomedical documents can be relevant in other cases as well. The enormous growth of information available to medical experts and medical researchers increases the demand for structured and compact information. Summarization of biomedical documents can be relevant in situations such as summarization of patient records.

1.2 Problem Statement

In the last decade, the field of natural language processing (NLP) has shown significant improvements. Research in this area is of great interest and with a very active research community, including many big tech companies such as Google, Microsoft, Facebook, and OpenAI. New solutions and improvements are published rapidly. Automatic text summarization is one of the popular downstream tasks in NLP. However, previous work has focused on single-document summarization, typically of news articles and web pages. It has, to the best of our knowledge, paid little attention to biomedical multi-document summarization.

A challenging problem that arises with biomedical text is that vocabulary and expressions are very different from the general domain. NLP techniques that are trained using general domain might not work well on biomedical text. Additionally, biomedical articles tend to be longer, and many natural language processing methods have limitations on the input size. The methods either do not accept long inputs or lack sufficient capacity to extract information from the whole input.

Further, multi-document summarization is a complex and challenging problem. The system must capture and manage redundant, complementary, and conflicting information to create a good summary. In addition, the amount of text data increases with the number of documents. There is limited literature on multi-document summarization of text from the biomedical domain, especially with the use of pretrained NLP models. In order to utilize the power of the very promising and recent techniques in NLP, adaptions are needed to create summaries efficiently.

Evaluating the performance on multi-document summarization models is not straightforward. To the best of our knowledge, there exists no dataset for evaluating biomedical multi-document summarization. With no such dataset, it is not easy to evaluate how adjustments affect our system and how it performs against other systems. We addressed this issue by combining two articles from our datasets, which contains PubMed articles, and using their concatenated abstracts as gold summary.

1.2.1 Research Questions

The main goal of our thesis is to investigate how text summarization and text mining techniques can be combined to generate biomedical multi-document summarization. As part of this, we propose a hybrid summarization model containing an extractive and an abstractive summarizer. We specifically focus on the extractive part of the system by experimenting with different techniques of representation, clustering, scoring, and selection of sentences. It is also desired that the summaries generated are concise and consistent. To ensure this, we will explore different evaluation methods. Based on this, the main problem addressed in this work can be expressed in the following main research question:

RQ: *How to generate multi-document summarization from biomedical texts using text summarization and text mining techniques?*

To be more specific, this main question can be divided into the following subquestions:

RQ1: *How can sentence embeddings capture semantics from biomedical texts?*

RQ2: *How can clustering, sentence scoring and sentence selection improve the process of extracting salient information?*

RQ3: *What evaluation methods can be used to verify that the summaries are non-redundant and preserve the most important information?*

1.2.2 Scope

The described summarization system is part of the larger In-Motion project in collaboration between St. Olav's University Hospital and the Norwegian University of Science and Technology (NTNU). We will not focus on the parts of the In-Motion system regarding the prediction of cerebral palsy and retrieval of documents associated with the prediction. Our main focus is on the summarization system alone, making it a system that is fully functional on its own, which could be integrated in the In-Motion system in the future. In addition, the system should be as fast as possible in order for it to be applicable in the real world. As this thesis is limited by both time and resources, we consider time and memory optimization of the text mining techniques used in the system to be beyond the scope of this thesis.

1.2.3 Contribution

For the explainability in the In-Motion system, the system must provide clarification of the CP predictions. As an explanation, a summary of relevant articles will support the decision-making.

The main contribution of this master thesis can be summarized as follows:

- We develop a hybrid multi-document summarization system for biomedical documents.
- We investigate what steps should be included in the processing flow and conduct an ablation study to determine what methods are best suited in the different steps.
- A dataset with CP-specific articles is constructed to evaluate the system further.
- The resulting system can support medical personnel to get a deeper insight into the In-Motion system's predictions.

1.3 Research Method

The research method used in this thesis is based on applying different solutions to the specified problem and evaluating them based on their performance. We initiated the thesis by collecting information about state-of-the-art methods within text summarization, focusing on articles related to the summarization of multiple documents and biomedical documents. The approaches were then evaluated based on different aspects, such as the techniques used and the applicability to our problem. Based on this, we constructed an ablation study plan consisting of the most promising subparts from the related systems. We created a dataset containing CP articles and selected the most suitable evaluation metrics for summarization tasks. The experiments related to the ablation study were conducted to find the optimal subpart of the system using the PubMed dataset. When the optimal processing flow was obtained, we evaluated the performance of two different pre-trained language models using both the PubMed and the CP dataset.

1.4 Thesis Structure

As mentioned earlier, parts of this thesis are obtained from our Specialization Project (Stang & Sollid, 2020). This is especially true for parts of Chapters 1 and 2. This thesis is structured as follows:

- **Chapter 1** introduces the motivation for this project and the challenges related to it, which is further defined through different research questions. A description of our research method is also included.
- **Chapter 2** contains the background theory that is relevant for the techniques used in our proposed system.
- **Chapter 3** gives an overview of previous work on automatic text summarization that is related to our system.
- **Chapter 4** describes our summarization system thoroughly and gives a detailed description of the experiments.
- **Chapter 5** presents the results obtained from the experiments that were conducted and the methods selected for each step of our processing flow.
- **Chapter 6** includes a discussion on the findings of our experiments and how the experiments were conducted.
- **Chapter 7** contains our conclusion of the thesis and our thoughts on future work for the project.

Chapter 2

Background

In this chapter, we present the theory that is relevant to our thesis. We start by giving an overview of artificial intelligence and some of its subfields, such as machine learning and deep learning. In addition, we look at the explainability of AI models. Further, methods within text mining and natural language processing relevant to our summarization system are addressed. Finally, we present the different approaches to automatic text summarization and how they can be evaluated. We chose to discuss these topics because they are relevant for the methods used in our proposed hybrid summarization system.

2.1 AI

Nilsson defines AI as following: "Artificial intelligence is a subpart of computer science, concerned with how to give computers the sophistication to act intelligently, and to do so in increasingly wider realms." (Nilsson, 1980) Nevertheless, defining intelligence is not easy. Alan Turing presented in 1950 the Turing test to provide an operational definition of intelligent behavior. A computer passes the test if a human interrogator cannot tell whether the conversation is with a computer or a human (Russell & Norvig, 2009). In order to imitate intelligent human behavior, a computer must possess many intricate capabilities. Fields such as machine learning, deep learning, and natural language processing are all under the umbrella of artificial intelligence, as shown in Figure 2.1.

2.2 Machine Learning

Machine learning is a field in AI where computers learn from experience and can act without being explicitly programmed. Already in 1968, Michie saw the possibilities of machine learning (Michie, 1968), but the past two decades have seen major

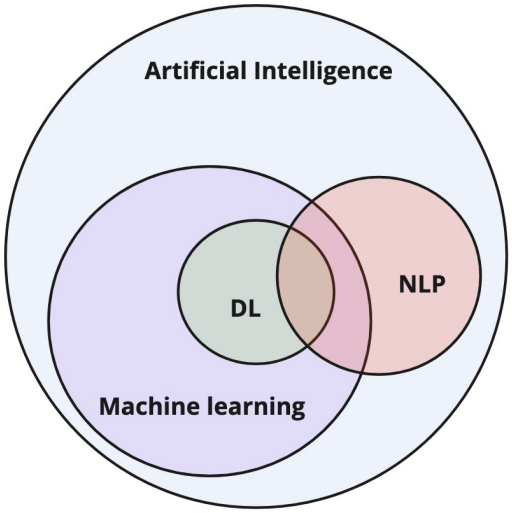


Figure 2.1: Machine learning, deep learning (DL) and natural language processing (NLP) are subfields of AI.

discoveries due to its popularity and access to increased computational power.

Supervised, unsupervised, and semi-supervised learning are common approaches to train a machine learning algorithm. Supervised learning is when the algorithm is fed example pairs of inputs and desired outputs, and the algorithm finds a way to produce the desired output based on the input. After this training phase, the algorithm will be able to create an output for an unseen input (James, 2018). In unsupervised learning, the output is unknown. The learning algorithm has to extract information from the input data. Typical unsupervised methods are clustering algorithms. Semi-supervised learning or self-supervised learning is similar to supervised learning, except for that the labels of training data are generated by the model itself (Goldberg, 2009). The model tries to predict one part of the input based on the remaining parts.

Machine learning uses mathematical and statistical theories to make models that recognize patterns. Conventional machine learning techniques require careful engineering in order to prepare the data into features that are understandable by the algorithm (Deng & Liu, 2018).

2.3 Deep learning

Deep learning is a subfield of machine learning that allows algorithms to learn representations directly from raw data (LeCun et al., 2015). The main concept is to automate the extraction of representation from the data (Najafabadi et al., 2015). Increasingly, more applications make use of deep learning techniques outperforming the previous state of the art machine learning methods. Until now, deep learning has achieved great success in computer vision, natural language processing, and speech recognition, but most likely, more fields will be added to the list (Najafabadi et al., 2015).

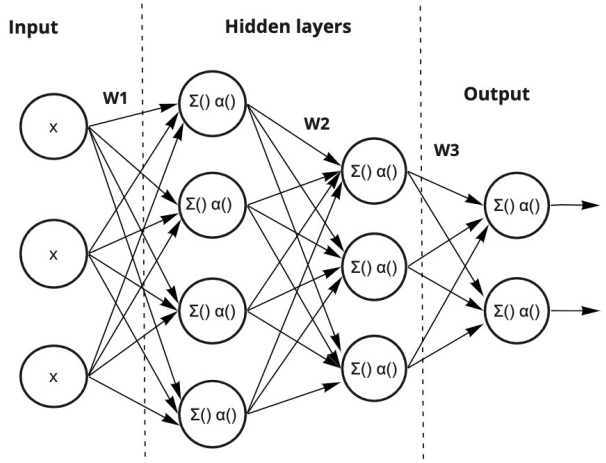


Figure 2.2: Fully connected multilayer perceptron (MLP) with two hidden layers.

A feed-forward network or multilayer perceptron (MLP) forms the basis of many deep learning models. The biological brain inspires the concept of neural networks; however, mathematics and statistics are the fundamentals. An MLP consists of an input layer, one or more hidden layers, and an output layer, where all layers are connected with weights to the adjacent layer (Sarkar et al., 2017). Figure 2.2 shows a fully connected MLP with two hidden layers. When the first hidden layer receives the input values from the input layer, it adds the values multiplied with its corresponding weight illustrated as $\Sigma()$ in Figure 2.2. The summed value is forwarded to the activation function, $\alpha()$ in Figure 2.2. The activation function's results are forwarded to the next layer, and the same procedure is done again. This is done until reaching the output layer, where the model outputs the classification or prediction.

For training the network, an algorithm called backpropagation is applied. When an example is fed through the network, a cost function computes the error between

the target output value and the calculated output value. The error propagates backwards, adjusting the weights in the layers. Repeating this over many examples will adjust the weights so that the network predicts as close to the target as possible.

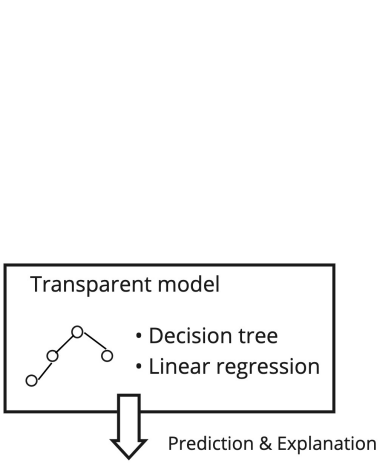
When a machine learning algorithm is "deep", it is often referred to as having more than one hidden layer in a neural network. Greater depth allows the network to learn more details and representation relationships within the data. Neural network models require less feature engineering making many time-consuming preprocessing steps in traditional machine learning obsolete. Furthermore, the same building blocks (i.e., layers) can be used in a variety of different tasks.

Despite the success of deep learning, there remain some challenges. Huge amounts of labeled training data and computational resources are required to train such a neural network. Also, the method lacks transparency and interpretability and is often regarded as black boxes. The complexity of the models makes them hard for a user to interpret the results. This has led to a new research area called Explainable AI.

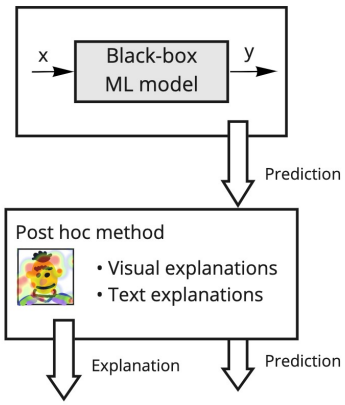
2.4 Explainable AI

Explainable Artificial Intelligence (XAI) is an emerging subfield of AI aiming to develop more transparent models that are more understandable to humans while maintaining high-performance levels (Adadi & Berrada, 2018). The lack of transparency and interpretability is a significant drawback in machine learning applications. Life-changing decisions such as a medical diagnosis needs explanation for both the medical expert and patient to trust the system. According to Adadi and Berrada, explanations of AI-based decisions are important to justify results, enhance control over vulnerabilities and flaws, iteratively improve models, and gain new knowledge. Also, the European Union introduced further initiatives to the field of XAI with GDPR (Goodman & Flaxman, 2017). From 2018 the law placed restrictions on automated individual decision-making that significantly affect users. As a result, a user has the right to receive an explanation of how the algorithm made the prediction and what data was affecting the outcome.

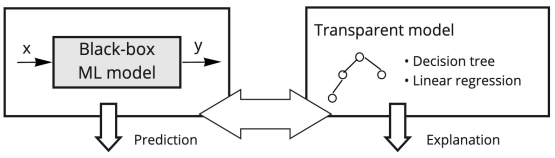
There are two main approaches in XAI; transparency-based and post-hoc (Dosilovic et al., 2018). Transparency-based XAI models, illustrated in Figure 2.3a, are when the model itself can explain the decision, limiting the model options to those with lower complexity. Simple models are easily understood and explain themselves, such as linear models or decision trees. In the family of transparency-based models, there also exists a hybrid approach, illustrated in Figure 2.3c, where a black-box model can be explained by associating it to a more interpretable and simple model. In the literature, it is often said that there is a trade-off between performance and transparency (Adadi & Berrada, 2018; Barredo Arrieta et al., 2020; Dosilovic et al., 2018). The more complex models, the more difficult to explain. Post-hoc methods,



(a) Simple and transparent method.



(b) Post-hoc method.



(c) Hybrid method.

Figure 2.3: Three approaches for XAI.

illustrated in Figure 2.3b, try to overcome this by keeping the complex machine learning algorithms and separately execute explanation techniques. The techniques are a kind of reverse engineering process that generates the explanation without knowing what is going on inside the black box. Thus the popularity of complex deep learning algorithms, the most recent works done in the XAI field, belong to post-hoc.

Techniques used to explain post-hoc try to enhance interpretability. We may distinguish among text explanations, visual explanations, local explanations, explanations by example, explanations by simplification, and feature relevance explanations (Barredo Arrieta et al., 2020). A popular visual method, called sensitivity analysis, is using a heatmap to show which pixels have been most relevant for the decision (Selvaraju et al., 2017). C. Yang et al. developed heatmaps for visually explaining CNN Alzheimer disease classification (C. Yang et al., 2018) and Papanastasopoulos et al. applied XAI visualization when classifying estrogen receptor status from breast MRI (Papanastasopoulos et al., 2020). Similar methods can be applied to text analysis, where important words can be highlighted in a visual explanation. On the other hand, there is limited research on text explanations of decisions; however, caption generation of videos and images is a more established research field (Bai & An, 2018; Dong et al., 2017; Hendricks et al., 2016).

2.5 Text Mining

Text mining is the process of extracting interesting and non-trivial patterns from unstructured text documents (A.-H. Tan et al., 1999). It includes several fields such as information retrieval, clustering, and summarization. Text mining usually involves structuring the data into better representations, deriving patterns, and evaluating the output.

Text representation The first step in text representation is to break down the text elements into meaningful tokens (Pinto et al., 2016). This process is called tokenization. Figure 2.4 shows a naive tokenization where the text is split by whitespace. However, tokenization can be more complex, e.g., identifying punctuation and subwords. Further, it is desirable to represent the text numerically in order to do mathematical operations. How to represent unstructured text numerically is one of the fundamental problems. A widely used text representation model is the Vector Space Model (VSM), where text documents are represented as numerical vectors (Yan, 2009).

Bag of words (BoW) is a commonly used VSM technique in traditional information retrieval (Yan, 2009). The whole set of terms in the text collection are considered as the vocabulary, except stopwords. The most straightforward BOW representation is the boolean model. A vector with the same dimension as the vocab-

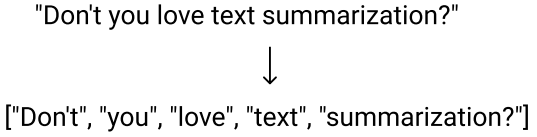


Figure 2.4: Tokenization where text is split by whitespace.

ulary represents a document. If a term is present in the document, there is a "1" in the term's position and "0" if absent (Yan, 2009). BoW does not consider word positions, and all words are considered equally important. Term Frequency-Inverse Document Frequency (TF-IDF) is an extension to BoW that aims to weigh the words in the vectors by their importance in the collection. TF-IDF is better than the boolean model but is not sufficient to capture the semantic meaning. In recent years, studies on neural vector representations on word-, sentence-, and document-level have emerged to overcome the BoW technique's drawbacks. Aiming to represent the text by considering semantic meaning, not only what terms are present.

Similarity measures Measuring similarity is necessary to organize and compare unlabeled documents into distinct groups. A similarity measure aims to evaluate the relationship between documents and give high scores to documents that contain the same information. Cosine similarity is one of the most popular measures when documents are represented as vectors (Allahyari et al., 2017b). Given two vectors $\vec{d}_1$ and $\vec{d}_2$ the cosine similarity is computed as follows:

$$cos(\vec{d}_1, \vec{d}_2) = \frac{\vec{d}_1 \cdot \vec{d}_2}{||\vec{d}_1|| \cdot ||\vec{d}_2||}, \tag{2.1}$$

where the numerator is the dot product between the two vectors, while the denominator represents the product of their Euclidean lengths (Schütze et al., 2008).

Another possible measure is the Euclidean distance. The Euclidean distance between two n-dimensional vectors x and y can be computed as follows:

$$d_{x,y} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.2}$$

Clustering Clustering is an unsupervised method that groups similar documents into coherent clusters. K-means is an iterative clustering algorithm, (P.-N. Tan et al., 2006) where K, the number of clusters, must be defined on forehand. It starts by partitioning the documents into K clusters by assigning a document to its closest initial centroid. Documents assigned to the same centroid form a cluster. The centroid value of each cluster is recomputed, usually based on the mean of the documents assigned. This process is repeated until the centroids converge.

Another clustering technique is Hierarchical Agglomerative Clustering (HAC) (P.-N. Tan et al., 2006). It starts with each document as a singleton cluster and then repeatedly merges the two closest clusters until all documents are in a single cluster. This type of clustering is often visualized using dendrograms to show the hierarchical relationships between the data points.

A challenge with clustering is to determine the number of clusters. A standard method for selecting k in K-means is the Elbow method (Kodinariya & Makwana, 2013). The Elbow method is a visual method where Within Cluster Sum of Squares (WCSS) is plotted for different numbers of k. For the first numbers of k, WCSS goes down rapidly. At one point, the WCSS begins to go down much slower. This is where the "elbow" is located, and the correct number of clusters is identified. In Section 2.5, the elbow is located at k=3. Another approach is to compute the average silhouette scores for a number of k's. The silhouette score aims to reflect the within-cluster tightness and separation between other clusters (Kodinariya & Makwana, 2013). The silhouette value ranges from -1 to 1, where a value close to -1 indicates that the entities are misplaced, and a value close to 1 implies that data is well clustered. If the value is around 0, it means that the entity could be placed in another cluster as well. When using silhouette scores for determining k, the k with the highest average silhouette score is selected. For clustering using HAC, it is possible to select the number of clusters based on its dendrogram.

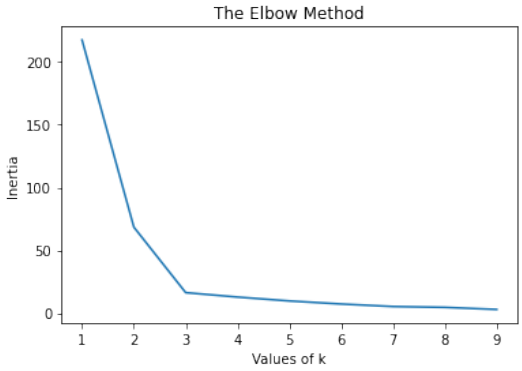


Figure 2.5: Plot of Within Cluster Sum of Squares of the inertias for different values of k.

2.6 NLP

Natural language processing (NLP) is a computational technique for automatic analysis and representation of human language. The field combines linguistics and artificial intelligence. NLP dates back to the 1950s with Alan Turing's Turing Test. Since then, NLP has aimed to facilitate interactions between computers and human languages (Deng & Liu, 2018). In the last two decades, machine learning ap-

proaches have dominated and become the foundation in NLP (Eisenstein, 2018). Further improvements were made when introducing deep learning. The NLP models were now capable of absorbing large amounts of training data. Typical NLP applications include speech recognition, machine translation, question answering, sentiment analysis, natural language generation, and text summarization.

The arrival of the RNN architecture was an important step in the use of deep learning in NLP. Cho et al. first proposed a model they called RNN Encoder-Decoder (Cho et al., 2014), and shortly after, Sutskever et al. presented their sequence-to-sequence model (Sutskever et al., 2014). Both are sharing the same idea of an encoder reading the input sequence. The encoder extracts the variable length input into a fixed-length vector representation that the decoder uses to generate the output sequence back into a variable-length sequence. Compared to the earlier approaches, this architecture's novelty is that the input and output length can vary from each other. The encoder and decoder both consist of either an LSTM unit or an RNN with a hidden unit inspired by LSTM (Hochreiter & Schmidhuber, 1997). LSTM is a recurrent neural network architecture with a memory cell capable of learning relatively long-term dependencies. One limitation of this architecture is when the fixed-length vector's dimension is too small for a long input sequence. In the next section, we will describe a mechanism called attention that was developed to overcome this challenge.

Attention Bahdanau et al. introduced attention in 2014 to overcome the bad memory in RNNs for the task of neural machine translation (Bahdanau et al., 2014). Attention is a trainable mechanism that captures complex dependencies between elements in a sequence. The technique is inspired by humans' visual attention, where the eyes can focus on one region with high resolution. With attention, the encoder-decoder has a better understanding of what is essential in the input sentence. The mechanism has been increasingly popular and has shown more use cases than only what it was introduced as, such as text classification, text summarization, and question answering.

The attention function computes a weight distribution on the input sequence, assigning higher values to more relevant elements; this is called the context-vector. Depending on the desired structure of the input and output data, the attention model varies. However, the core idea is the same; highlight the essential parts of the text.

Attention has emerged in recent years as a promising technology in natural language processing. Hu (2020) provides a review of current work on attention mechanisms. Since 2014, when attention was first introduced, the mechanisms have been further developed and become more complex. Different variants have been proposed, such as basic attention, multi-dimensional attention, hierarchical attention, self-attention, memory-based attention, and task-specific attention.

Self-attention is a variant of attention that is only based on the input sequence. It captures information about a word based on the position to other words in the

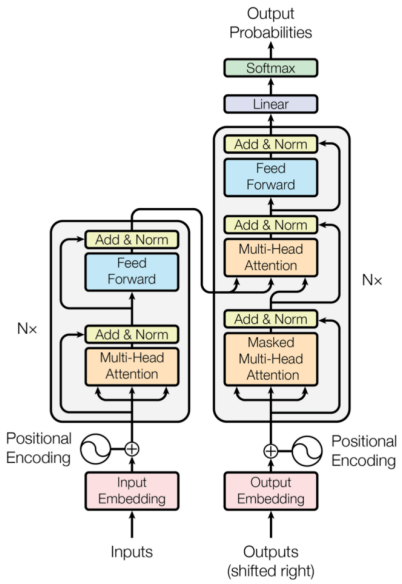


Figure 2.6: Illustration of Transformer architecture. (Vaswani et al., 2017)

sentence. The main advantage of this is that the model can attend information from different representations subspaces at other positions.

Another attention structure is hierarchical attention. Z. Yang et al., 2016 presents a hierarchical attention network for document classification with two levels of attention mechanism, both on word- and sentence level. Due to this, the model manages to extract important information globally and locally.

Attention was introduced as a supplement to RNNs, but the next section shows how attention redeemed RNNs.

Transformers In the paper "Attention is all you need", Vaswani et al. (2017) presented an encoder-decoder architecture independent of RNNs. The architecture was called Transformer. With the Transformers multi-head attention and positional encoding, there was no longer a need for RNNs and LSTMs. The architecture solely relies on the attention mechanism to extract global dependencies between inputs and outputs.

The Transformer uses an encoder-decoder design, see Figure 2.6 for illustration. In short, the encoder is fed an input sequence $x = (x_1, ..., x_n)$, and maps it to a continuous representation $z = (z_1, ..., z_n)$. Further, the decoder generates an output sequence $y = (y_1, ..., y_m)$ based on z.

In order to fully understand the Transformer architecture, it is necessary to describe its building blocks. An encoder-layer consists of two sublayers; a multi-head self-attention mechanism and a fully connected feed-forward network. Both are followed by a layer normalization with a residual connection, meaning $LayerNorm(x+$

$Sublayer(x)$). The decoder has the same architecture as the encoder but with an additional sublayer that performs multi-head attention over the encoder's output and masks the output embedding. The masking layer will hide all words after the word the decoder is trying to predict, letting it only know what is already "written." The novelty with the Transformer was to exploit the information stored in the attention context vectors. The multi-head attention consists of several scaled dot-product attention functions. Positional encoding is required to represent the order of a sequence when there is no recurrence in the model. A majority of the pre-trained models developed after the release of Transformers use this architecture or with some modifications.

Pre-trained Language Models

Language modeling is one of the core components in modern NLP (Qiu et al., 2020). It involves analyzing enormous amounts of text data in order to determine the word probability. In other words, the language model learns the probability with which a sequence of words will follow each other (Deshpande, 2020). The training of general-purpose language models, using large amounts of unannotated data, is known as pre-training. Pre-training helps the model reason about the different characteristics and structure of general language.

The pre-trained word representations can be non-contextual or contextual. The models using non-contextual representations create a single word embedding representation for each word in their vocabulary. On the opposite, models using contextual representations generate a word representation based on the remaining words in the sentence. The contextual models can use either unidirectional or bidirectional representations (Devlin & Chang, 2018).

Once the language model is pre-trained, it can be utilized for any downstream tasks, such as text summarization and question answering. This utilization is beneficial, as many task-specific datasets contain very little data. Using the pre-trained models as a foundation for learning task-specific models helps overcome the data limitation and avoids the need for training a new model from scratch (Gu et al., 2020).

Figure 2.7 illustrates the pre-training and fine-tuning of neural language models. The first step involves training the model on massive amounts of unannotated data. Then, a smaller task-specific dataset is fed into the model, fine-tuning it and making it capable of performing the intended downstream task on a test dataset.

Types of pre-trained models Pre-trained language models can be divided into three different categories, depending on their usage of the transformer architecture. The models can be autoregressive, autoencoding, or sequence-to-sequence.

Models using an autoregressive objective use only the decoder part of the ori-

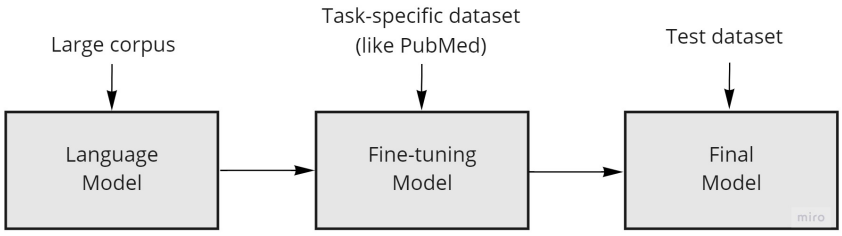


Figure 2.7: The process of training a language model.

ginal transformer (Vaswani et al., 2017). In addition, they use an attention mask, so the models are able to see the tokens before the attention heads at each position (Z. Yang et al., 2019). However, they are not able to see the tokens after. The pre-training of autoregressive models is based on the classic language modeling task; having read all previous tokens, guess the next one. The unidirectionality makes the autoregressive models most suited for tasks like text generation.

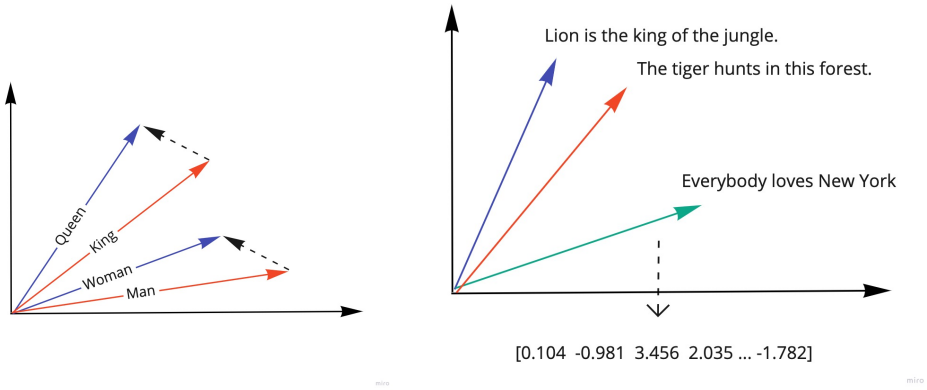
As with the autoregressive models, the autoencoding models use only the decoder part of the original transformer. However, they do not use attention masks, resulting in the model being able to see all the tokens in the attention heads (Z. Yang et al., 2019). The models are pre-trained by corrupting the input sequence before they try to reconstruct the original sequence. The bidirectionality of autoencoding models makes them applicable to many tasks, such as text generation or sentence/token classification.

Models based on the sequence-to-sequence objective rely on both the encoder and the decoder of the original transformer. A masked sequence is fed into the encoder before the decoder sequentially produces the masked tokens in an autoregressive way (Qiu et al., 2020). These models can be fine-tuned for tasks like translation, question answering, and summarization.

Word and Sentence Embeddings

Text representation is an important part of text summarization and text mining techniques. Machine learning algorithms often require the input text to be fixed-length, and the choice of representation can impact the success of the method. The first proposal of distributed representation of words came in 1986 by Rumelhart (Rumelhart et al., 1986). Even though the problem of representing words as vectors is old, many new contributions have been made to the field after the introduction of encoder-decoder-based word embedding techniques. Embeddings encode words and sentences, and this can drastically improve data processing.

Word embeddings Word embeddings involve representing words as real-valued vectors. Semantically similar words will have a similar representation, i.e., they will



(a) Word embeddings. **(b)** Sentence embeddings.

Figure 2.8: Words and sentences can be represented by vectors, which are often called embeddings.

be close to each other in the vector space. A common example is "King - Man + Woman = Queen", illustrated in Figure 2.8a. The two most used word embeddings are Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). Both models use an unsupervised training objective and are based on the assumption that words occurring in the same context have a tendency to have similar meanings. Word2Vec comes in two different versions; Continuous Bag-of-Words (C-BOW) and Skip-Gram. The goal of CBOW is to predict a target word based on its neighboring words, ignoring the ordering of the words. As opposed, Skip-Gram selects a word and uses this to predict its neighboring words. GloVe (Global Vectors) learns word embeddings by looking at how frequently words appear together in a corpus.

Sentence embeddings As a result of word embeddings' success, the research has expanded to representing longer text strings. As with word embeddings, sentence embeddings involve representing a sentence as a dense fixed-length continuous vector and can be used for understanding the context of the words. This is illustrated in Figure 2.8b. Sentence embeddings can be divided into traditional approaches and neural approaches. A baseline in traditional approaches involves representing each sentence as a Bag-of-Words, using a word embedding such as Word2Vec, and then averaging the word vectors. This approach does not take the ordering of the words into account.

Neural approaches involve pre-training a model on large text corpora. It has become a well-studied field, and several methods have been introduced. SkipThought (Kiros et al., 2015) uses an unsupervised training objective to train an RNN-based encoder-decoder model. The model tries to predict the neighboring sentences from the current sentence. InferSent (Conneau et al., 2018), on the other hand, uses a supervised training objective to learn universal sentence embeddings. The model

consists of a siamese bi-directional LSTM network trained on the Standford Natural Language Inference (SNLI) dataset (Bowman et al., 2015). Universal Sentence Encoder (Cer et al., 2018) is a multi-task learner as it expands unsupervised learning with training on the labeled SNLI dataset. It can be seen as a generalization of the InferSent and the SkipThought models. The embeddings created by the model are specifically targeted to handle transfer learning to other NLP tasks. Neural embeddings have achieved state-of-the-art results in several NLP tasks and have become an essential part of modern NLP methods.

Embeddings in biomedicine The amount of information that is available in the biomedical domain is increasing fast. This results in an increased need for NLP techniques to help retrieve and analyze the data. When using text mining techniques on biomedical and clinical text, it is critical that the sentence semantics are well captured. Traditional methods or neural methods pre-trained on general domain might not model biomedical information accurately due to natural language ambiguity and can suffer from the out-of-domain issue (Chen et al., 2019). Both word embeddings and sentence embeddings have been adapted to biomedical and clinical data (Chen et al., 2019; Chiu et al., 2016; Pyysalo et al., 2013; Th et al., 2015), in order to overcome the problems mentioned.

2.7 Text Summarization

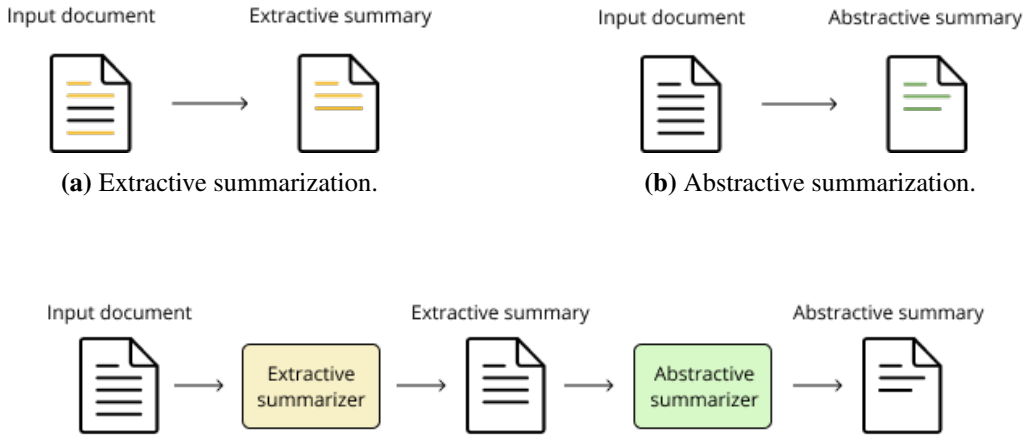
According to Radev et al. (2002), a text summary is "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that.". This definition points to three critical aspects concerning text summarization; the summary should be short and preserve relevant information from single or multiple documents.

Automatic Text Summarization

Automatic text summarization is an area under NLP that involves creating concise and coherent summaries without human interaction. As the number of available documents has increased tremendously, comprehensive research has been required. Several techniques and methods for automatically summarizing text have been developed, and the application of these methods spans different domains, including the biomedical (Allahyari et al., 2017a). Here, automatic text summarization can be used to summarize medical documents, reducing the time needed for doctors to read through articles searching for information. In addition to decreasing reading time, automatic summarization can help decision-making and increase the number of documents processed by a person (Zheng et al., 2020).

However, there are many challenges regarding automatic text summaries. Creating summaries comparable to human-created summaries is difficult, as computers lack human knowledge and language capability (Allahyari et al., 2017a). Also, an automatic text summarization solution needs to ensure that the summary information is reliable. It is critical for many downstream tasks that the summary is accurate and effectively covers the text's semantically relevant aspects.

There are mainly three different approaches to automatic text summarization; extractive, abstractive, and hybrid approaches.



(a) Extractive summarization.

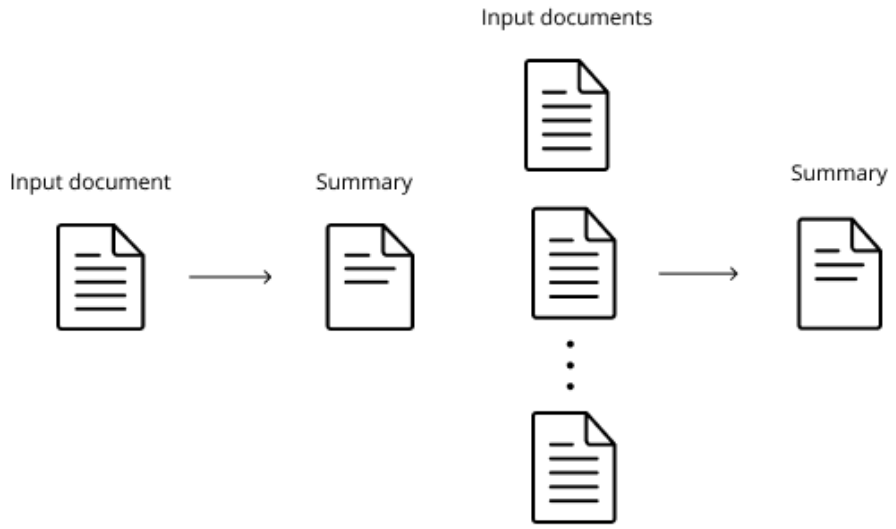
(b) Abstractive summarization.

(c) Hybrid summarization.

Figure 2.9: Illustrations of extractive, abstractive and hybrid summarization.

Extractive Extractive summarization, illustrated in Figure 2.9a, selects the most important sentences from the input text and concatenates the sentences in their entirety to form the final summary (El-Kassas et al., 2021). Typical steps in an extractive summarization system include representing the sentences, giving them a score estimating the importance, and extract the top K sentences. An extractive approach can efficiently generate a summary; however, information redundancy and incoherence between summary sentences are apparent drawbacks.

Abstractive Abstractive summarization, shown in Figure 2.9b, aims to capture the main content and generate new concise sentences resulting in a fluent and condensed summary (Hou et al., 2018). Generating sentences is done in a word-by-word manner, possibly with words never used in the original texts. An abstractive approach's main advantage is that they are more similar to human-written summaries. However, generating high-quality abstractive summaries is a complicated task, especially concerning semantics and natural language (Hou et al., 2018). The majority of the state-of-the-art abstractive summarization techniques use Transformer based encoder-decoders to create summaries (Zheng et al., 2020). A shortcoming



(a) Single-document summarization. **(b)** Multi-document summarization.

Figure 2.10: Illustrations of single-document summarization and multi-document summarization.

with the Transformer architecture is that computational costs are quadratic to the input length.

Hybrid A third approach is a two-phased hybrid approach, illustrated in Figure 2.9c. It combines extractive and abstractive summarization. The typical architecture first performs extractive summarization to select important sentences and then use them as input to an abstractive summarization model (El-Kassas et al., 2021). Hybrid approaches are popular when dealing with long or multiple documents (P. J. Liu et al., 2018; Subramanian et al., 2019). The motivation for using a hybrid approach is to use the efficient extractive approach to reduce the input text before using the more computationally expensive model to generate an abstractive summary. Research has also shown that compressing the input with a content selection step before performing an abstractive step improved the summaries (F. Liu & Liu, 2009; Mehdad et al., 2014; Subramanian et al., 2019).

Automatic text summarization can also be divided in how many documents are summarized. We divide it into two categories; single-document summarization and multi-document summarization, as illustrated in Figures 2.10a and 2.10b. Both approaches aim to compress the text to a summary that contains the most important information, but they require individual adjustments.

Single-document summarization Single-document summarization produces a summary generated from a single document. Abstractive summarization methods have shown great improvements on SDS the recent years. NLP models can achieve human performance on summarization tasks with high-linguistic quality on the summaries (Zhang et al., 2019). In the literature, there has been an extensive focus on summarizing news and other shorter texts. However, there is a rising interest in summarizing long documents like scientific articles (Zhang et al., 2019). Most existing pre-trained models do not have the capacity for documents longer than 512 or 1024 tokens. Lengthy documents can contain much noise, so capturing the document's essence can be more difficult. The naive approach is to truncate the documents only considering the beginning of the document, but this can lead to the loss of important information as the main subjects might be widely scattered over the text. It is also possible to prioritize the document parts that are most likely to contain the essential information, thereby only needing to summarize parts of the document, as done in Gidiotis and Tsoumakas.

Multi-document summarization Multi-document summarization is generated from multiple topic-related documents (Widyassari et al., 2020). It is considered more complicated than single-document summarization since the multiple documents can contain more redundant, complementary, and conflicting information. Also, the amount of text data that needs to be compressed is larger, which leads to higher computational complexity. A multi-document summarization systems' goal should be to generate summaries that are non-redundant, cover the information about all topics in the documents, and the information included in the summary should be relevant for the reader.

Prior work has focused on extractive methods, but recently abstractive methods with neural pre-trained models have been applied to multi-document summarization (W. Li et al., 2020). The abstractive methods for multi-document are complex and still have a limitation on the amount of input. Hybrid approaches have shown good performance when summarizing multiple long documents (P. J. Liu et al., 2018).

There are two ways of concatenating multiple documents; flat or hierarchical concatenation. Flat concatenation is the simplest approach where all documents are merged into a flat sequence of text. The difficulty is that the models need the ability to process long sequences and discover redundancy in the flat text. Hierarchical concatenation process the documents with cross-relation in mind. The most popular hierarchical method is on word/sentence-level using clustering algorithms or graph-based techniques to capture cross-document relations (Ma et al., 2020).

Summarization Evaluation

Evaluating text summaries is a massive challenge as there is no optimal metric for comparing different summary approaches. Additionally, most documents or sets of

documents have no ideal summary to compare with the generated summary (Das & Martins, 2007).

The simplest and possibly most accurate approach for summary evaluation is using humans to evaluate the quality. This approach involves humans judging different quality metrics like content, conciseness, coherence, grammaticality, and readability. However, this is extremely expensive with respect to time, and it is challenging to conduct frequently.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) was introduced by Lin (2004) and is one of the most used evaluation metrics. It is a set of evaluation metrics that automatically determine a summary's quality by comparing it to other human-made summaries. ROUGE bases itself on counting the number of overlapping units, such as n-gram, word pairs, or word sequences. The overlap of units is also known as recall. Recall is the proportion of words in the reference summary that are also present in the computed summary. Precision, on the other hand, is the proportion of words in the computed summary that are also in the reference summary.

The ROUGE measures are recall-based as they look at the overlap between a constructed and a gold standard summary.

- ROUGE-N is a comparison of n-grams.
    - ROUGE-1 considers the overlap of unigrams (each word)
    - ROUGE-2 considers the overlap of bigrams (every two consecutive words)
- ROUGE-L considers the longest common subsequences (LCS)

Even though ROUGE is the most used evaluation metric for NLP tasks like summarization, it has its flaws. As ROUGE only measures word overlap, it is possible to achieve high ROUGE scores for a poorly written summary. Another drawback with ROUGE is that it requires a gold standard summary to compare with the candidate summary. Creating these human-written summaries is an expensive process.

Chapter 3

Related Work

This chapter presents some of the systems in the literature that are related to our research. We are aware that numerous other relevant systems exist, but we selected those that we found especially relevant and will focus on them. We divide the systems into extractive, abstractive, and hybrid summarization models.

3.1 Extractive Summarization Models

LexRank

LexRank is a much-mentioned algorithm in the literature and is often used as a baseline in multi-document summarization systems (Erkan & Radev, 2004). LexRank constructs a graph by creating a vertex for each sentence in the documents. The edges between the vertexes represent the cosine similarity between the TF-IDF vector representations of the sentences. Further, the sentences are ranked inspired by PageRank, aiming to find the most central sentences (Page et al., 1999). The ranking follows a voting mechanism where central sentences give higher weighted votes to similar sentences. To get a high score, a sentence must be similar to many sentences that are in turns also similar to many other sentences. A summary is formed by combing the top k central sentences using a threshold or output length limit.

CIBS

Clustering and Itemset mining based Biomedical Summarizer (CIBS) is a multi-document summarization system (Moradi, 2018). It exploits itemset mining and Unified Medical Language System (UMLS) (Nelson et al., 2001) to summarize biomedical documents. UMLS is a thesaurus of biomedical concepts that allow the translation of noun phrases from the input text to concepts. The itemsets of concepts extracted represent a sentence. Further, the system applies frequent itemset mining

on the concepts to extract the main subtopics. A hierarchical clustering algorithm divides the sentences into multiple clusters where sentences in the same cluster cover the same topics. The summary is produced by selecting the sentences that cover most topics in each cluster. Due to the lack of a biomedical multi-document summarization dataset, the author (Moradi) constructs a dataset. With a disease name as a query, the first 300 abstracts were retrieved from PubMed. The gold summary to the collection was provided by the Wikipedia article of the same disease. This was repeated for 25 diseases and constituted the dataset. The paper states that CIBS can perform better than other comparison methods and produce more informative and related summaries.

SoBA

In 2020 Moradi published another article on biomedical extractive text summarization (Moradi, Dashti et al., 2020), this time a single-document system with the use of word embeddings and graph ranking. Due to convenience, we name the system with the title's acronym, SoBA. The input text was modeled as a weighted, undirected graph where the relatedness of sentences was computed with cosine similarity between the vector representations from the word embeddings. In the experiments conducted, different word-embeddings and graph ranking algorithms were compared. The authors tested three well-known word representations, Word2Vec's SkipGram and CBOW, and GloVe, which they all trained on a large corpus of biomedical texts. Additionally, BioBERT's pre-trained contextual word representations were tested. They experimented with combinations of context-sensitive and context-free embeddings and found that when GloVe-embeddings complement BioBERT's contextualized embeddings, the system can represent semantic relations and context of sentences more accurately than with only one embedding type. PageRank, HITS, and PPF were tested as graph ranking algorithms where PageRank gave the best results.

SummPip

SummPip is a multi-document summarization system that converts documents into a sentence graph, clusters the graphs, and applies cluster sentence compression to summarize (J. Zhao et al., 2020). SummPip represents sentences with the use of word embeddings. J. Zhao et al. employ a naive approach by taking the mean of word vectors from Word2Vec. The graph is built with linguistic knowledge metrics and cosine similarity between the sentence representation vectors. Further spectral clustering is applied on the Laplacian matrix computed from the sentence similarity graph. The last step in the SummPip pipeline is multi-sentence compression. A single summary sentence is generated for every k clusters, combing key phrases from different sentences in the cluster. The final summary consists of key phrases from the original text, unlike other typical extractive approaches that ex-

tract whole sentences. The system achieves competitive results when comparing ROUGE scores, but the summaries are less fluent and more redundant than manual gold summaries.

SciSummPip, a single-document system inspired by Summpip, tests two sentence embeddings in addition to Word2Vec (Ju et al., 2020). These being SentenceBERT and SciBERT embeddings. SciBERT is a BERT-model pre-trained on scientific texts, while SentenceBERT is a modification of BERT that is trained to find similar sentences in vector space. The domain-specific SciBERT gave best results. However, SentenceBERT had a competitive performance with significantly less workload.

ExMEmb

Lamsiyah et al. present ExMEmb, named with title's acronyms for convenience. ExMEmb is an extractive centroid-based multi-document summarization system that utilizes sentence embeddings and selects relevant sentences based on three scores (Lamsiyah et al., 2020). These being; content relevance score, novelty score, and position score. Additionally, an empirical analysis of nine sentence embeddings models was conducted.

First, the input sentences are embedded with a sentence embedding model. Next, the centroid vector is computed from the mean of all sentence vectors and is further used to compute the relevance score for each sentence. The novelty score and position score are also computed for each sentence before the three scores are combined. Finally, the top-ranked sentences are selected for the extractive summary. The top 5 embeddings models are uSIF (Ethayarajh, 2018), USE-DAN, USE-Transformer (Cer et al., 2018), NNLM (Bengio et al., 2003), and the InferSent-GloVe (Conneau et al., 2018). The system was evaluated on DUC'2002-2004 and outperformed other centroid-based methods and achieved promising performance compared to recent deep learning-based methods.

3.2 Abstractive Summarization Models

GraphSum

GraphSum is an end-to-end neural-based model that leverages graph structures to capture cross-document relations. It produces abstractive summaries from multiple documents. They introduce a graph-informed attention mechanism that incorporates graphs into the document encoding process. The graph structure is also utilized in the summary generation with a hierarchical graph attention mechanism. The model is trained on general-domain. However, it is possible to combine it with other pre-trained models, necessitating a costly pre-training step. The paper states that the

model can extract salient information from long documents and generate coherent summaries more efficiently.

3.3 Hybrid Summarization Models

GeWiS

P. J. Liu et al. proposed in 2018 a multi-document summarization system that recreates English Wikipedia articles from cited source documents and Web Search results on the topic (P. J. Liu et al., 2018). We name the system GeWiS for convenience. In the constructed dataset, the order of magnitude to the input and output sizes are $10^2 - 10^6$ and $10^1 - 10^3$ words, respectively. To overcome the very large input size, a two-staged extractive-abstractive approach is needed. A subset of the original input is selected with an extractive approach, while a transformer decoder is used to generate the summary. Paragraphs are ranked using TF-IDF computations as in a query retrieval problem where the query is the article's title. The top-ranked paragraphs, sorted with the most relevant in the beginning, are input to the decoder. For the generating step P. J. Liu et al. utilize a modified decoder inspired by the Transformer architecture (Vaswani et al., 2017). The combination of extractive and abstractive approaches appears to significantly affect the final performance compared to approaches using only one approach when summarizing long documents.

SEAL

Segment-wise Extractive Abstractive is a long document transformer-based summarizer, but the approach is also applicable to flat concatenated multiple documents (Y. Zhao et al., 2020). Input documents are divided into sequences of snippets. Further, Y. Zhao et al. study four approaches for handling long inputs; Truncation, Compressive-Abstractive, Extractive-Abstractive, and SEAL. Truncation cuts the input document to the maximum input length of the transformer. Compressive-Abstractive compresses the snippets to shorter representations and concatenates the shorter representations as to the decoder input. The Extractive-Abstractive (EA) approach encodes the snippets separately, assigns scores, and selects snippets to feed the transformer decoder. SEAL encodes the snippets similarly to EA but utilizes a segment-wise scorer to better select snippets. Of the approaches tested, SEAL performs best. Also, it achieves state-of-the-art performance on the datasets ArXiv and PubMed. Unlike other hybrid models, SEAL train the extractive and abstractive stage jointly.

ExAbSum

Subramanian et al. proposed a long-document summarization system that we name ExAbSum, utilizing transformer language models in a hybrid approach. The system is built for summarization of scientific articles. For the extractive phase, a hierarchical seq2seq sentence pointer with an LSTM encoder, with word and sentence level LSTMs, is used to point out sentences. The abstractive phase consists of a transformer architecture identical to GPT-2 (Radford et al., 2019). This required an extensive pre-training step. The summaries are conditioned on the introduction of the original article and the extracted sentences.

Subramanian et al.'s method outperforms several previous extractive and abstractive summarization methods on ArXiv and Pubmed datasets. They also focus on *abstractiveness*, meaning that the model does not generate summaries that contain copied phrases or sentences.

Tretyak aims to improve the system above by using a pre-trained model instead of training it from scratch. The paper experimented with BERT (Devlin et al., 2018), ROBERTA (Y. Liu et al., 2019) and ELECTRA (Clark et al., 2020) for the extractive summary. The pre-trained autoregressive models BART and GPT-2 are tested for the abstractive stage. BERT generated the best extractive summary, while BART conditioned on the introduction, extractive summary, and conclusion, in that order, gave the best overall ROUGE scores. Removing the extractive step leads to a decreased ROUGE score.

CAiRE

With the rapid increase in articles concerning COVID-19 research and the urgent need for insights on the pandemic, the Allen Institute for AI, among others, created the COVID-19 Open Research Dataset (CORD-19).[1] The aim is to facilitate the development of data mining and text mining tools that can help the medical community. With this challenge in mind, Dan et al. created CAiRE COVID, a neural-based question answering and query-focused multi-document summarization system. It was awarded as the winner of one of the CORD-19 Kaggle challenges.

CAiRE COVID is a system that combines QA techniques and summarization techniques for mining available biomedical literature. More specifically, the system consists of three main parts:

1. Document Retriever
2. Relevant Snippet Selector
3. Multi-Document Summarizer

The first two parts of CAiRE COVID form an open-domain question answering system. The Document Retriever pre-processes the query by paraphrasing it into

[1] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge

shorter queries that are easier for the system to handle. The updated queries are then inserted into the Snippet Selector. The snippet selector consists of two QA modules; the HLTC-MRQA (D. Su et al., 2019) and the domain specific model BioBERT (Lee et al., 2020). It returns the n most relevant paragraphs containing the most relevant answer snippets from the retrieved articles. The paragraphs are re-ranked, and answers are highlighted. The last part of the system generates both an abstractive and an extractive query-focused summary. The abstractive model is based on BART (Lewis et al., 2019), conditioned on the top-n ranked paragraphs from the QA system, the predicted answers snippets, and the query itself. The abstractive summary can be classified as a hybrid approach since selected parts are fed to the model. The extractive model re-rank sentences from the paragraphs according to a query relevance score to form a summary. The sentences and the query are represented by the average of ALBERT's contextualized embeddings (Lan et al., 2019). The three sentences with the highest cosine similarity score to the query are chosen for the summary. The system's final result is two concise summaries and a ranked list of relevant paragraphs from a given query's retrieved documents. CAiRE is the only system of our knowledge that implements a hybrid approach on biomedical documents. However, the system is QA specific. In addition, the system is dependent on a starting question, and the resulting summaries are very short.

3.4 Summary

The different systems above all describe summarization systems and are in different ways related to our work. In Figure 3.1 we build a taxonomy that categorizes the systems described in Chapter 3.

CIBS, SoBA, and CAiRE summarize biomedical articles, where CIBS and CAiRE also are multi-document summarization. Further, Lexrank, SummPip, GraphSum, and GeWiS are general-domain multi-document systems. GeWiS, SEAL, ExAbSumm, and CAiRE describe hybrid systems to manage a large amount of input text.

Prior work shows that when summarizing biomedical articles, methods adjusted to the biomedical domain perform better than general models. We also experienced this in our specialization project, where Pegasus (Zhang et al., 2019) pretrained on Pubmed articles performed best on summarizing single biomedical articles (Stang & Sollid, 2020).

Several studies have been done on multi-document summarization, but the community has not agreed on its optimal approach, and the problem is solved in many different ways. An abstractive approach gives more fluent summaries than an extractive. However, an end-to-end abstractive multi-document summarization system, like GraphSum, requires complex architecture and a costly pre-training step. In order to utilize the powerful Transformers for abstractive summarization, a pre-

processing stage can be conducted. Extractive summarization techniques are used to select a subset of sentences to feed an abstractive model. A hybrid approach has the potential for summarizing long or multiple articles.

With the rise of neural word and sentence embeddings, systems like SoBA and SummPip utilize this technique to represent sentences in the extractive summarization approach. Moradi improved his results with SoBA and got better results than CIBS, indicating that representing sentences with word- or sentence embeddings is a promising technique. To overcome the redundancy of information in multi-document summarization, CIBS and SummPip integrate clustering techniques to group topics and choose sentences from different clusters.

In order to solve our challenges, we explore combining various methods from the state-of-the-art. Our system is inspired by several of the systems presented in this chapter. We think a hybrid multi-document summarization system is promising due to the massive amounts of input the system needs to tackle. Further, SoBA, ExMEmb, CAiRE, and SummPip show the potential for the use of sentence embeddings. From CIBS and SummPip, clustering seems to be an essential step for detecting topics in the documents. LexRank and ExMEmb assign scores to the sentences and rank their importance based on this. Besides the QA-focused system CAiRE, no one has constructed a hybrid system to suit multi-document summarization of biomedical documents. A more extensive description of the processing steps we include in our system and the methods we experiment with are described in the next chapter.

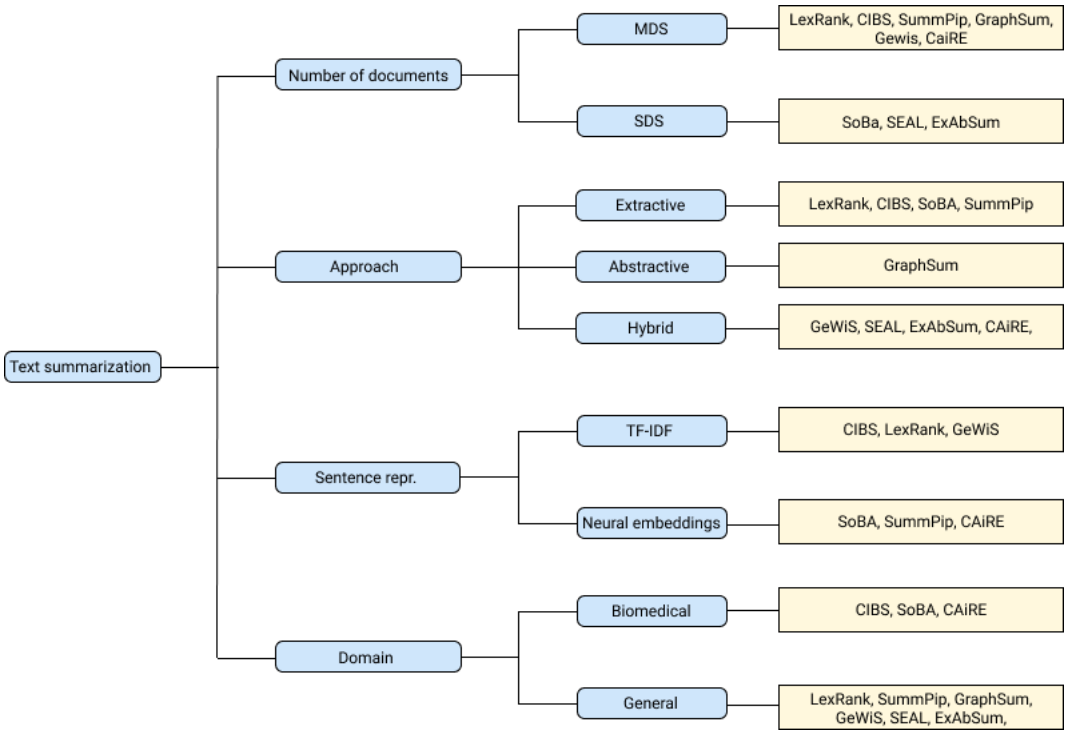


Figure 3.1: Taxonomy of related text summarization systems.

Chapter 4

Approach

We begin this chapter by describing our processing flow, presented in Section 4.1. In Section 4.2, we give an overview of our ablation study, while in Section 4.3 we present each method that is tested out thoroughly. Finally, the datasets and evaluation metrics used for our experiments are presented in Section 4.4.

4.1 Processing Flow

We propose a multi-document summarization system that can enhance the explainability of the In-Motion application. A considerable amount of time was spent investigating what techniques could be used to develop a promising summarization system that will consider the different challenges with multi-document summarization of biomedical articles. The processing flow is based on a review of related work as well as results from our specialization project (Stang & Sollid, 2020). We start by providing an overview presenting the system's steps and proceed to describe our considerations of each step. What methods we experiment with and end up using at each step will be presented in Section 4.2 and Chapter 5, respectively.

The processing flow consists of 9 steps. As illustrated in Figure 4.1, the system (1) reads in multiple biomedical documents, (2) preprocess the documents and divide into sentences, (3) compute the sentence embeddings, (4) cluster the sentence embeddings, (5) give the sentences scores, (6) select sentences based on their scores, (7) concatenate sentences to an extractive summary, which is further (8) fed to a pre-trained model that generates (9) the final abstractive summary.

A preprocessing stage is necessary before computing the sentence embeddings in order to ensure the desired format on the input text. Text representation can be done in different ways. TF-IDF representations are a traditional method used in text mining and extractive summarization (Moradi, 2018) but do not capture the similarity between words and the semantic structure of sentences. As described in Chapter 3, neural sentence embeddings is a promising technique to represent sen-

tences. According to (Jiang et al., 2020), neural embeddings are better at capturing semantic similarity between sentences than TF-IDF-based approaches and performed better on document summarization tasks. Many neural sentence embedding models are pre-trained to capture the similarity between sentences making them a good option when comparing sentence similarities. We therefore apply neural sentence embeddings in our processing flow to represent sentences.

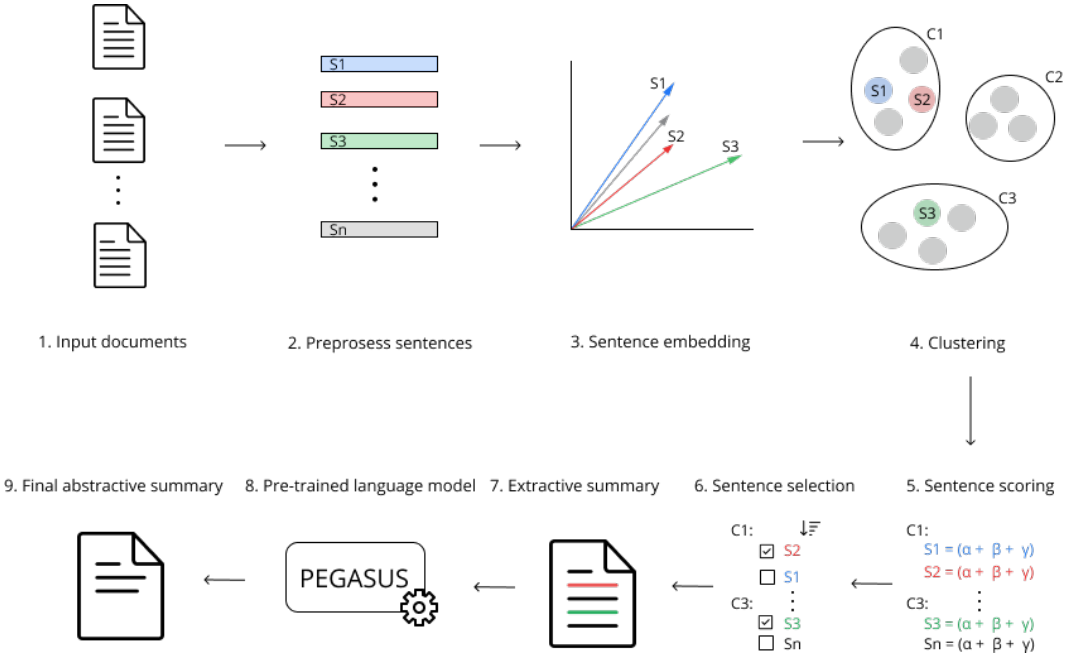


Figure 4.1: The processing flow to produce the summarization.

The motivation for doing a clustering step is to avoid similar sentences in the summary. In multi-document summarization, the documents may cover many of the same topics. By clustering group topics, we can carefully select sentences with different topics. Several studies, such as Alguliyev et al. (2019), Moradi (2018) and J. Zhao et al. (2020), have used clustering in their text summarization systems.

Now that sentences are grouped in clusters, we need a method to select sentences. The scoring step assigns scores to the sentences based on different metrics. The scoring method aims to give high scores to the most important sentences from the cluster. Further, sentence selection determines how many sentences from each cluster will form the extractive summary.

There are two options with the desire for abstractive final summaries in multi-document summarization: an end-to-end abstractive system or a hybrid approach. A hybrid approach combines extractive summarization techniques with an abstractive model. An end-to-end abstractive system requires advanced natural language understanding and generation, which often consists of deep neural networks requiring extensive pre-training steps. Even though the number of publications on deep learning multi-document summarization models has increased rapidly in the recent years

(Ma et al., 2020), they are complex and difficult to customize to biomedical articles. For that reason, we decided on a hybrid approach to utilize the powers of the pre-trained models for single document summarization while making adjustments for multiple biomedical documents input.

4.2 Ablation study

In our experiments, we conduct an ablation study. "An ablation study is a scientific examination of a machine learning system by removing its building blocks in order to gain insight on their effects on its overall performance" (Sheikholeslami, 2019). The experimental field of neuropsychology has inspired ablation studies in machine learning. In the 19th century, a physician removed parts of pigeon brains to study how it affected the pigeon's behavior (Yildirim & Sarikcioglu, 2007). When developing a neural network or other complex algorithms or systems, ablation studies allow the researchers to identify where the performance improvements come from.

In our case, the system will collapse if we remove some steps entirely from the pipeline. Instead, we substitute a step with other simpler approaches. This kind of ablation study is called a substitution study (Cohen & Howe, 1988). The steps in the pipeline we experiment with are sentence embedding, clustering, sentence scoring, and sentence selection. At each step, we have several alternatives, including a naive approach, except for the sentence selection step, where we have no naive approach. For each step, the alternative with the best performance is retained. By performing an ablation study, we can observe how much a feature contributes to the performance without running all possible combinations of a model. The ablation study plan is illustrated in Figure 4.2.

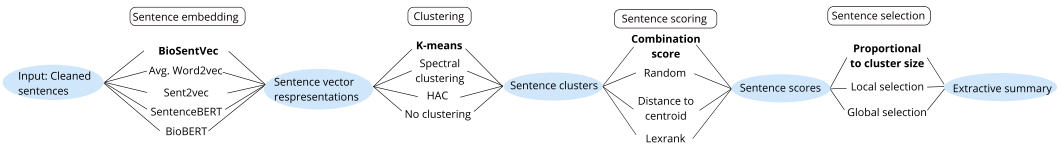


Figure 4.2: Illustration of our ablation study. Approaches in bold constitute the base pipeline.

Before performing the experiments in the ablation study, we decided on a base for our pipeline. The steps in this pipeline are used in the experiments until we obtain results from the ablation study. Each step consists of the approach we believe is most promising. For this pipeline, we select BioSentVec, K-means clustering with cosine similarity, combination score, and proportional sentence selection.

4.3 Summarization Pipeline

4.3.1 Preprocessing

The first part of our summarization pipeline involves preprocessing the documents. We start by sentence tokenizing and word tokenizing each document, using the NLTK tokenize library.[1] Further, stopwords and punctuations are removed from the text, similarly like in BioSentVec (Chen et al., 2019). We use the NLTK English stopwords corpus for the removal of stopwords.[2] Punctuations are removed using Python's string library.[3] Finally, sentences containing less than three or more than 100 tokens are removed.

4.3.2 Sentence Representation

A way to convert sentences into vector representations is a necessary step in the pipeline to apply machine learning techniques and compute similarities in the proceeding steps. The sentence embeddings presented below are the ones used in the experiments. These includes Sent2Vec (Pagliardini et al., 2018), BioSentVec (Chen et al., 2019), SentenceBERT (Devlin et al., 2018), and BioBERT (Lee et al., 2020). Sent2Vec is a general-purpose sentence embedding approach, while BioSentVec is a domain-specific version of Sent2Vec, trained on biomedical articles. BioBERT is a domain-specific pre-trained model with BERT architecture. SentenceBERT is a modification of BERT and the current state-of-the-art approach in sentence embeddings. In addition to these three, we use averaged Word2Vec, described in Section 2.6, as a naive sentence embedding approach.

SentenceBERT

The introduction of BERT (Devlin et al., 2018) set a new state-of-the-art performance on sentence-pair regression tasks. However, BERT's architecture makes it unsuitable for unsupervised tasks such as clustering and semantic similarity search. In order to overcome this limitation, SentenceBERT (Reimers & Gurevych, 2019), which is a modification of the pre-trained BERT network, was created. By fine-tuning BERT using siamese and triplet network structures (Schroff et al., 2015), SentenceBERT is able to obtain sentence embeddings. Through the fine-tuning, it is also able to ensure that it maintains BERT's accuracy. SentenceBERT is fine-tuned on Natural Language Inference (NLI) data and evaluated on common transfer learning tasks and Semantic Textual Similarity (STS) tasks. For almost all evaluation tasks, the sentence embeddings derived from SentenceBERT achieve state-of-

[1] https://www.nltk.org/_modules/nltk/tokenize.html
[2] http://www.nltk.org/nltk_data/
[3] https://docs.python.org/3/library/string.html

the-art results and significantly outperform other methods such as Universal Sentence Encoder (Cer et al., 2018) and InferSent (Conneau et al., 2018). In our implementation, SentenceBERT is implemented using the SentenceTransformers Python framework [4]. The BERT model applied in the experiments is the *paraphrase-distilroberta-base-v1*, a DistilBERT-base-uncased model.

BioBERT

BioBERT (Lee et al., 2020) is a domain-specific pre-trained language model for biomedical text mining. It uses the architecture of BERT (Devlin et al., 2018) and the same WordPiece tokenization vocabulary. BioBERT is initialized with BERT's weights and further pre-trained on PubMed abstracts and PubMed Central full-text articles. Since BioBERT is using the same general-domain vocabulary as BERT, there are many out-of-vocabulary words. However, since WordPiece is a subword tokenizer, the unknown words are represented with frequent subwords. With domain-specific pre-training, BioBERT outperformed BERT on several biomedical text mining tasks.

Sentence embeddings can be obtained from BioBERT by averaging output layers. However, Reimers and Gurevych (2019) and B. Li et al. (2020) show that averaging BERT embeddings get lower performance on STS tasks. Despite these results, we wanted to test an in-domain BERT model to generate sentence embeddings. Additionally, SciSummPip, mentioned in Section 3.1, achieved the best results when averaging SciBERT layers compared with Word2Vec and SentenceBERT.

In our implementation of BioBERT embeddings, we utilize the BERT implementation from HuggingFace [5] with BioBERT pre-trained weights [6] from (Lee et al., 2020). A sentence is tokenized into n tokens and fed to the model. We extract the first and the last hidden layer with dimensions (1, n, 768) from the model's output. Following J. Su et al. (2021), we take the average of the first and last layer since it achieved better results than only taking the last layer. Next, in step 1 in Figure 4.3, the layers are averaged over the n tokens. Then, in step 2 in Figure 4.3, we average the first and the last layer to finally obtain the sentence embedding with dimension (1, 768).

Sent2Vec

Sent2Vec is a model for creating sentence embeddings (Pagliardini et al., 2018), and it is based on the Continuous Bag-of-Words (CBOW) model. In order to train sentence embeddings instead of word embeddings, Sent2Vec adapts CBOW's unsupervised training objective. It creates sentence embeddings by combining word

[4] https://github.com/UKPLab/sentence-transformers
[5] https://huggingface.co/transformers/model_doc/bert.html
[6] https://huggingface.co/dmis-lab/biobert-v1.1

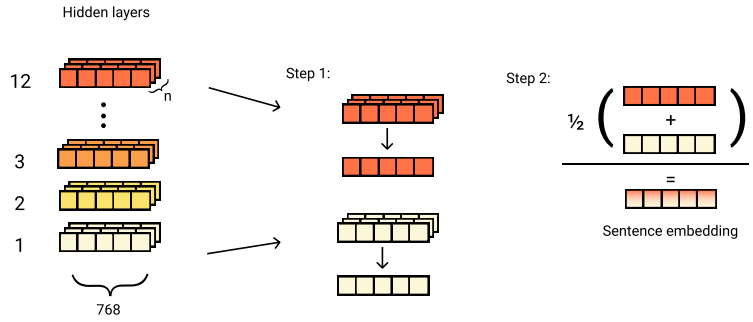


Figure 4.3: Method of obtaining sentence embedding from BioBERT.

vectors with n-gram embeddings. Sent2Vec is a simple and computationally efficient model, with a computational complexity for the embeddings of O(1) vector operations for each word that is processed. The training of Sent2Vec was done using different datasets; the Toronto book corpus and Wikipedia sentences and tweets. For the evaluation of Sent2Vec, a standard set of unsupervised and supervised benchmark tasks, such as the Semantic Textual Similarity benchmark, was used. Cosine similarity was used to compute the sentence similarity. The results from the experiments that were conducted showed that Sent2Vec achieves state-of-the-art performance on most tasks. The outperformance of other sentence embeddings indicates that Sent2Vec's general-purpose embeddings are robust and applicable for many different tasks. For the implementation of Sent2Vec, we employ the Sent2Vec library [7] using the *sent2vec_bigrams* model which is pre-trained on English Wikipedia pages.

BioSentVec

BioSentVec is a domain-specific sentence embeddings model (Chen et al., 2019). The sentence embeddings are created by training the more general-purpose Sent2Vec model (Pagliardini et al., 2018) on domain-specific data. BioSentVec was trained using more than 30 million documents from the PubMed dataset and the MIMIC III dataset. By pre-training BioSentVec on in-domain, it is able to obtain sentence embeddings that are robust and generalizable on various text genres in the clinical domains and in biomedicine. The evaluation of BioSentVec was done on two different tasks, being sentence similarity and multi-label text classification. The datasets used for the first experiment were BIOSSES (PubMed articles) and MedSTS (clinical notes), while for the second experiment, they used the Hallmarks of Cancer corpus. In both cases, the usage of BioSentVec gave the highest performance compared to other methods. The results from BioSentVec indicate that the embeddings are better at capturing sentence semantics than other embeddings such as Doc2Vec and Universal Sentence Encoder. BioSentVec is implemented using the same lib-

[7] https://github.com/epfml/sent2vec

rary as Sent2Vec, but the Sent2Vec model is replaced with the *BioSentVec_bigram* model that is pre-trained on both PubMed articles and MIMIC III clinical notes.

4.3.3 Clustering

For the clustering step, we compare three different clustering techniques as well as no clustering. We want to observe what contribution clustering has on the resulting summary and potential differences in the applied clustering techniques. In addition, we run the clustering with two differed measures; cosine similarity and Euclidean distance. Since the sentence embeddings are evaluated on cosine similarity measures, we found it reasonable to base the algorithms on this. However, Moradi, Dorffner et al. (2020) got slightly better results with Euclidean measures when deriving embeddings by averaging BERT's output. Therefore, we also experiment with using Euclidean distance.

We test three different clustering algorithms. The selection of algorithms is based on what others in the literature use. K-means is the most popular method and can be found in various summarization systems (Alguliyev et al., 2019; Haider et al., 2020; Miller, 2019; Waheeb et al., 2020). The use of Hierarchical Agglomerative Clustering (HAC) is inspired by Moradi. He uses HAC in his summarization systems (Moradi, 2018; Moradi, Dorffner et al., 2020), while spectral clustering is applied by Summpip (J. Zhao et al., 2020) and also SciSummpip (Ju et al., 2020).

Selecting Number of Clusters

Selecting the optimal number of k is not a straightforward approach. First, we tried the Elbow method. We plotted the WCSS of the inertias for k ranging from 3 to 10, but there was no clear elbow. Next, we calculated the silhouette scores for the different k's, but all k's gave a score between 0.10 and 0.15. The marginal differences were not sufficient to determine k. Since neither the Elbow method nor the Silhouette method gave any significant results for selecting k, the ROUGE scores for the different k's were compared. For k ranging from 3 to 10, we summarized 100 PubMed articles. Figure 4.4 shows the plot of the k's with the resulting ROUGE scores when using K-means with cosine similarity as the measure. Since all three ROUGE scores were slightly higher on k=8, we used 8 clusters in our implementation.

K-means

K-means is a traditional partitioning algorithm, described in more detail in Section 2.5. We use skicit-learn[8] for implementation of K-means with Euclidean distance. When running with cosine similarity, we use an implementation from NLTK

[8]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

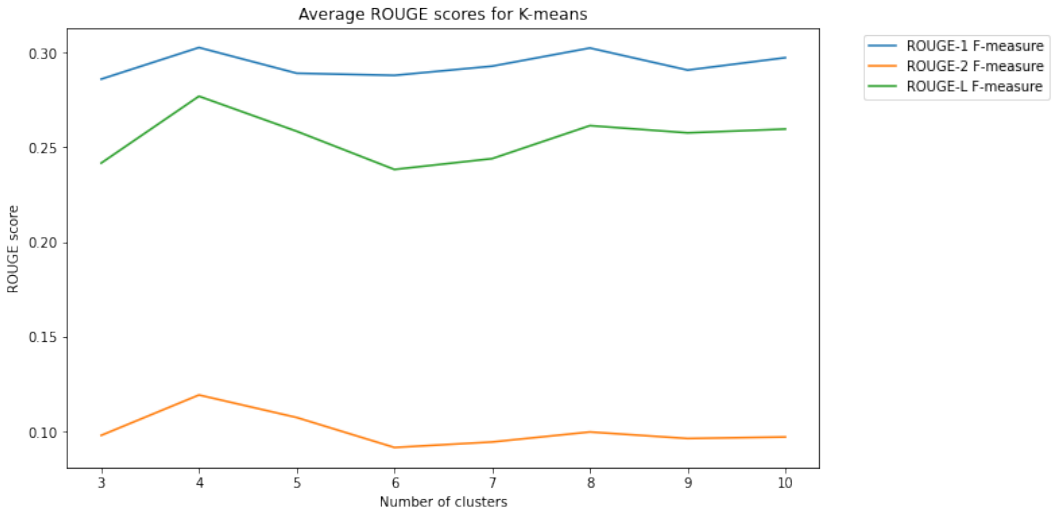


Figure 4.4: Average ROUGE scores for different number of clusters when using K-means with cosine similarity on 100 PubMed articles.

[9]. For both implementations, we use a method from Scikit-learn that initializes centroids to be distant from each other to converge faster.

HAC

Hierarchical Agglomerative Clustering is also described in Section 2.5 and is a clustering algorithm where all sentences start as individual clusters, and for each step, the closest clusters merge (P.-N. Tan et al., 2006). We apply complete linkage as proximity for our implementation of cosine, meaning that the two clusters with the smallest distance between the two farthest points are merged. For the Euclidean implementation, we apply Ward's linkage as proximity. Ward's method merges the two clusters with the smallest sum of squares from the centroid to the data points in the clusters. We implement HAC with Skicit-learn [10] with both measures.

Spectral Clustering

Spectral clustering often outperforms traditional clustering as K-means and can be solved efficiently with standard linear algebra software (Von Luxburg, 2007). The first step is to compute a similarity matrix between the objects, which can also be seen as a fully connected graph with a similarity measure as edge weights. In our case, we compute the cosine similarity matrix between the sentences. Next, the first K eigenvectors of the Laplacian matrix are computed to define a feature vector for

[9]https://www.nltk.org/api/nltk.cluster.html
[10]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

each sentence. Then, the sentences are clustered with K-means. Spectral clustering is implemented with Scikit-learn.[11]

4.3.4 Sentence Scoring

Scoring of sentences is an essential part of extractive text summarization and determines which sentences will be extracted to form the summary. The sentences with the highest scores are the ones that are selected. Therefore, the scoring system should score sentences by giving high scores to the most relevant sentences in the text and lower scores to the less relevant ones. In addition, the score should help avoid redundancy in the obtained summary. The sentence scoring approaches used for the experiments are random scoring, distance to centroid, LexRank, and a combination score. Random scoring is a naive approach and does not consider the content of a sentence. Both distance to centroid and LexRank are traditional approaches for scoring sentences, but they do not consider redundancy. The combination score is a more complex approach that, in addition to scoring the sentences based on their content, also tries to avoid redundancy.

Random Scoring

The random scoring approach involves assigning a random number to each sentence inside a cluster. The random number is a floating-point number between 0 and 1 and is generated using the Python function *random.random()*.[12] This approach results in a selection of random sentences for the extractive summary.

Distance to Centroid

Another approach for scoring the sentences inside a cluster is by computing the sentences distance to the cluster centroid. The centroid is computed by averaging the vector embeddings for the sentences that belong to the cluster. The scoring of each sentence is performed by computing the cosine similarity (Equation (2.1)) between the sentence vector and the centroid vector. This approach results in assigning high scores to the sentences that capture the essence of the cluster.

LexRank

LexRank is described in Section 3.1, but there the entire summarization system is described. For sentence scoring in our system, only parts of the system are employed. More precisely, we want to apply the computation of centrality that includes the voting mechanism. In our implementation, we utilize methods used in the

[11] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html
[12] https://docs.python.org/3/library/random.html

LexRank Python package[13]. First, we compute the cosine similarity matrix between the sentence embeddings. Next, the matrix is fed to a method that computes each sentence's centrality in the cluster. As a result, each sentence is assigned a score that can be sorted to retrieve the most important sentences.

Combination Score

The combination score approach is inspired by Lamsiyah et al.'s article (Lamsiyah et al., 2020). The article combines three different sentence scoring metrics: content relevance score, novelty score, and position score. The primary purpose of the content relevance score and position score is to find sentences that capture the essence of the input document in a good way, while the novelty score helps reduce redundancy. This approach achieved good results and outperformed similar methods. However, their experiments were only tested on news articles. In a study conducted by Plaza and Albornoz, they evaluate different position scoring approaches on biomedical articles (Plaza & Albornoz, 2012). They conclude that the type of position score used by Lamsiyah et al. does not work that well on biomedical articles, as the article structure is very different from news articles. Instead, they discovered that scoring sentences based on the article section it is positioned in gave good results. This approach requires all articles to be in the same format with the same sections. We will only use the content relevance score and novelty score and not any positional information. In this way, we will obtain a more general-purpose system that is able to process all kinds of articles, regardless of their structure.

Content relevance score The content relevance score for a sentence is computed as the distance between the sentence and the cluster centroid, described in Equation (4.1). It is the same score as the "Distance to centroid" score presented previously. The content relevance score is a floating-point number bounded in [0, 1]. Sentences that are assigned a higher content relevance score are considered more relevant, and sentences assigned a lower score are considered less relevant.

$$score^{contentRelevance}(S_i, D) = cosineSimilarity(\overrightarrow{S_i^D}, \overrightarrow{C_D}) \tag{4.1}$$

Here, S_i represents the sentence belonging to cluster D. $\overrightarrow{S_i^D}$ is the vector embedding of sentence S_i and $\overrightarrow{C_D}$ is the vector embedding of cluster C_D.

Novelty score The novelty score of a sentence is used for reducing redundancy. Novel sentences are assigned a high score, while sentences that are very similar to other sentences are penalized and assigned a lower score. The computation of the novelty score is inspired by Joshi et al. and is described in Equation (4.2) and

[13]https://github.com/crabcamp/lexrank

Equation (4.3). We start by computing the cosine similarity for each sentence in a cluster to all other sentences inside the cluster.

$$sim(S_i, S_k) = cosineSimilarity(\overrightarrow{S_i^D}, \overrightarrow{S_k^D}), 1 \leq k \leq N, i \neq k \tag{4.2}$$

Here, S_i and S_k are sentences belonging to cluster D, and $\overrightarrow{S_i^D}$ and $\overrightarrow{S_k^D}$ are their vector embeddings, respectively. N is the total number of sentences in cluster D.

The novelty score $score^{novelty}$ is then computed using Equation (4.3). A threshold τ is used to determine if a sentence is novel or redundant. If the maximum of the obtained similarities $max(sim(S_i, S_k))$ for a sentence S_i is below the threshold, it is considered novel. If it is above the threshold but has a higher content relevance score compared to its most similar sentence, it will be assigned a high novelty score. The novelty score is a floating-point number in the range [0, 1].

$score^{novelty}(S_i, D)$

$$= \begin{cases} 1, \text{ if } max(sim(S_i, S_k)) < \tau,\ 1 \leq k \leq N,\ i \neq k \\ 1, \text{ if } max(sim(S_i, S_k)) > \tau \text{ and} \\ \qquad score^{contentRelevance}(S_i, D) > score^{contentRelevance}(S_l, D), \\ \qquad l = \arg\max(sim(S_i, S_k)),\ 1 \leq k \leq N,\ i \neq k \\ 1 - max(sim(S_i, S_k)), \text{ otherwise} \end{cases} \tag{4.3}$$

Here, l is the index for the sentence in cluster D that is most similar to sentence S_i. N is the total number of sentences in cluster D. We use the same threshold, $\tau = 0.95$, as Lamsiyah et al. for the computation of the novelty score, as their empirical research concluded with this being the optimal threshold value.

The resulting score, $score^{combination}$, is a combination of the content relevance score and the novelty score.

$$score^{combination}(S_i, D) = \alpha * score^{contentRelevance}(S_i, D) + \beta * score^{novelty}(S_i, D) \tag{4.4}$$

Here, $\alpha + \beta = 1$. We base our values for α and β on the values obtained by Lamsiyah et al., since the optimal values for these parameters have been tested thoroughly. These values are $\alpha = 0.6$, $\beta = 0.2$ and $\sigma = 0.2$, for the content relevance, novelty and position scores, respectively. Since we only compute the content relevance and novelty scores, we adjust the values to suit our experiment, while still preserving the relationship between the parameters. Therefore, we set the values to $\alpha = 0.75$ and $\beta = 0.25$.

4.3.5 Sentence Selection

We evaluate three different approaches for selecting sentences for the extractive summary. These being local selection, proportional selection, and global selection. Sentences are selected based on the score they are assigned and the number of sentences the extractive summary should contain. We select the number of sentences in the extractive summary based on the maximum input length of the pre-trained model, which for Pegasus is 1024 tokens. First, we use 100 documents from our dataset to estimate the average number of words in a sentence. Here we preprocess the dataset by removing punctuations but not stopwords, as we use whole sentences as input for the pre-trained model. We get an average of 26 words per sentence. Since Pegasus can accept 1024 tokens, this gives us $1024 \div 26 \approx 40$ sentences in our extractive summary. Before the sentences are fed to Pegasus, the sentences are sorted based on their score, placing the best scored sentences at the beginning of the extractive summary.

Local Selection

Selecting sentences using the local selection strategy involves choosing the k highest scored sentences from each cluster. k is computed using Equation (4.5). This approach results in an equal contribution to the extractive summary from all the generated clusters. Large clusters are assumed to represent the main topics of the input document, while small clusters represent less important ones. Since small clusters have the same contribution as large clusters, this may result in unnecessary information being included in the summary. If there are clusters with less than k sentences, the cluster will contribute with fewer sentences, and the extractive summary will be shorter.

$$k = \left\lceil \frac{N}{C} \right\rceil \tag{4.5}$$

Here, N is the number of sentences that the extractive summary should contain, and C is the number of clusters. We use $ceiling$ to ensure that the maximal input length of the language model is obtained.

Proportional Selection

The proportional selection approach involves selecting sentences from a cluster based on the size of the cluster, i.e., how many sentences it contains. This approach is based on the assumption that large clusters most likely represent the key topics of the input document, and therefore should have a larger contribution to the extractive summary. In this way, all clusters contribute to the summary while ensuring that the summary captures the essence of the documents. Equation (4.6) is used for computing the number of sentences k_i that cluster C_i contributes with to the summary.

$$k_i = \left\lceil \frac{N * C_i}{M} \right\rceil \tag{4.6}$$

Here, N is the number of sentences that the extractive summary should contain, C is the size of the n-th clusters, and M is the total number of sentences in the input document.

Global Selection

The global selection selects the N highest scored sentences across all clusters, where N is the total number of sentences to select to the extractive summary. With this approach, not all clusters need to contribute to the extractive summary, only the clusters with high-scored sentences.

4.3.6 Abstractive Step

Recent years have seen a rise in pre-trained models that perform abstractive summarization. Pegasus has received state-of-the-art performance and gave the best results in the experiments conducted in our *Specialization Project* (Stang & Sollid, 2020). Many of the existing transformer-based language models, including Pegasus, use a full self-attention mechanism in their language models. This attention mechanism has a computational and memory requirement that is quadratic to the sequence length. For NLP tasks that require longer input sequences, like biomedical document summarization, the quadratic dependency reduces the applicability. BigBird-Pegasus, with its sparse attention mechanism, achieves state-of-the-art on long document summarization. In the ablation study, we use Pegasus to generate abstractive summaries, as the implementation of BigBird-Pegasus was not yet released when we began our experiments. However, BigBird-Pegasus was added to the Huggingface Transformers library on May 7, 2021. Therefore, we perform some experiments using BigBird-Pegasus for the abstractive summarization to see if it can improve our pipeline. After the ablation study was completed, we experimented with modifying the Pegasus model parameters and tested feeding both Pegasus and BigBird-Pegasus with different input lengths.

Pegasus

Pegasus (Zhang et al., 2019) is a sequence-to-sequence model for abstractive summarization developed by Google. It uses the standard Transformer encoder-decoder architecture introduced in (Vaswani et al., 2017). The novelty of Pegasus' architecture is its self-supervised pre-training objective. The researchers experimented with the choice of pre-training corpus, pre-training objective, and vocabulary size to obtain the best outcomes.

Pegasus uses a new technique in the pre-training objective; Gap Sentence Generation (GSG), specially adapted for summarization. Whole sentences are selected and removed from the input text and concatenated into a target summary. The removed sentences are replaced with a mask before each of them is reproduced using the remaining sentences. To choose which sentences to remove, Pegasus uses a technique they call Principal selection. The top m scored sentences, based on the ROUGE-1 scores for the sentences and the input document, are removed. Zhang et al. aim to remove the most important sentences. The idea is that using a pre-training objective close to the task of summarization will lead to better and faster fine-tuning performance.

Pegasus is pre-trained on C4 and HugeNews, which are web crawler text and news text, respectively. Further, Pegasus is fine-tuned on 12 downstream datasets, including PubMed. They observed that Pegasus managed to perform surprisingly well with only 1000 fine-tuning examples. As with other pre-trained models, Pegasus has a maximum input length of 1024 tokens. The model achieves state-of-the-art results on several datasets for single-document summarization.

We utilize HuggingFace's Transformers library[14]to download the Pegasus model with the PubMed fine-tuned checkpoints. If the extractive summary from the previous stage exceeds the limit of 1024 tokens, Pegasus truncates the input and bases the summary on the first 1024 tokens.

BigBird

In July 2020, Google researchers published "Big Bird: Transformers for longer sequences" (Zaheer et al., 2020). BigBird has a sparse attention mechanism that reduces the dependency from quadratic to linear while at the same time preserving the properties of the full-attention mechanism.

For the development of BigBird, graph sparsification methods were used as inspiration. In each layer of the transformer, a generalized attention mechanism is applied (Zaheer et al., 2020). It is described in the form of a directed graph, where the edges represent the set of the inner products that the attention mechanism will consider. The sparse attention mechanism combines three attention mechanisms; random attention, window attention, and global attention.

BigBird is set up on two different models; ROBERTA (Y. Liu et al., 2019) and Pegasus (Zhang et al., 2019). The ROBERTA version is trained for tasks like QA and classification, while Pegasus is trained for encoder-decoder tasks like summarization. BigBird-Pegasus is trained on PubMed, ArXiv, and BigPatent, which are datasets that contain long documents. The sparse attention mechanism is only applied on the encoder side while keeping full attention on the decoder side since, in summarization, the input length is typically long while the output length is small.

The results of BigBird show that the sparse attention mechanism is just as powerful and expressive as the full self-attention mechanism. The results hold for

when it is used in a standalone encoder, as well as in an encoder-decoder transformer. Compared to previously possible lengths of input sequences, BigBird can handle four times the length of Pegasus' limit. BigBird achieves state-of-the-art results for several NLP tasks, including document summarization and question answering. It shows that modeling a longer context encoder gives significantly increased results.

For our implementation of BigBird, we utilize the same library as with Pegasus.[14] We use the BigBird-Pegasus with checkpoints fine-tuned on Pubmed.

4.4 Evaluation

4.4.1 Dataset

PubMed dataset PubMed is a search engine accessing biomedical and life sciences literature from mainly the extensive database MEDLINE. In the experiments, we use a dataset presented in a paper by Cohan et al. retrieved from PubMed Open-Access repositories (Cohan et al., 2018). Every article is stored to make it possible to access the abstract, article body, and different sections separately. The dataset consists of 133k articles. However, we use a small subset consisting of 150 documents in the ablation study and 1000 documents in a final evaluation experiment.

CP dataset Since the InMotion project concentrates on the early prediction of cerebral palsy, we constructed a small dataset consisting of 72 articles related to cerebral palsy. We utilized an API provided by NIH[15] for accessing the PubMed database. First, we tried using the *ESearch* endpoint with cerebral palsy keywords as a query but experienced retrieving many articles that were not about cerebral palsy, missing the whole point of creating our own dataset.

After conversations with the physician Lars Adde at St. Olavs hospital, he recommended a review article on early diagnosis of cerebral palsy (Novak et al., 2017). With this article as a starting point, we use the *ELink* endpoint, which looks up similar, related, or otherwise connected records in the same database to the provided article ID. We provide to the endpoint the ID of the article (Novak et al., 2017) and retrieve a list with the linked articles' IDs. Next, we use the *EFetch* endpoint to retrieve the full-text articles. The articles are provided in an XML format requiring a preprocessing step to extract the article text while leaving other elements behind, like tables and citations. Each CP article is stored with the abstract and article text separated.

[14] https://huggingface.co/transformers

[15] https://dataguide.nlm.nih.gov/eutilities/utilities.html

For the experiments in the ablation study, we use the PubMed dataset as it contains more documents than the CP dataset. In the absence of a biomedical multi-document dataset, the experiments are based on summarizing two biomedical articles. As input to the summarization pipeline, the two article texts are concatenated. Similarly, the two abstracts of the articles are concatenated and serve as the reference summary for the evaluation. Every experiment runs the summarization system for the 150 first articles from the dataset, resulting in 75 multi-document summaries. As a final validation of our summarization system, we test both Pegasus and BigBird-Pegasus using the CP dataset.

4.4.2 Evaluation Metrics

The experiments conducted in the ablation study are evaluated using the ROUGE-metric presented in Section 2.7. The generated summary is compared to the reference summary using the F-measure of ROUGE-1, ROUGE-2, and ROUGE-L. The ROUGE scores are implemented using Python's ROUGE library.[16]

Measuring redundancy is not a straightforward approach. As of what we know, there is no standard method of evaluating redundancy in summaries. We, therefore, perform an informal evaluation where we evaluate redundancy in the generated summaries. In lack of time and resources, we perform the evaluation ourselves. We consider repeated sentences or phrases and do not look at the medical synonyms and facts as we have no medical background. We define a summary as highly redundant if four or more sentences are excessive in the summary, moderate to little redundant if there are one to three sentences that are excessive, and non-redundant if no sentences are redundant.

[16] https://pypi.org/project/rouge/

Chapter 5

Results

In this chapter, we describe the results obtained from our experiments. Section 5.1 presents the ROUGE F-measure scores for the different approaches in each step of the pipeline, as described in Chapter 4. In Section 5.1.5, we present the final pipeline based on the results from Section 5.1. In Section 5.2, the results from the different experiments with the both Pegasus and BigBird on the two different datasets are presented. Lastly, Section 5.3 shows the informal evaluation of redundancy in the generated summaries. All of the experiments in this project are carried out on NVIDIA Tesla V100 GPU or NVIDIA Tesla P100 GPU.

5.1 Ablation Study

This section contains the results from the ablation study. We present the average of the ROUGE scores for the 75 abstractive multi-document summaries generated by the summarization pipeline. As described in Section 2.7, an abstractive summary generates new sentences with new word combinations for the summary. Therefore, we emphasize the ROUGE-1 scores more than the ROUGE-2 and ROUGE-L scores.

5.1.1 Sentence Embeddings

The first part of the ablation study was to select the optimal sentence embedding approach for the summarization pipeline. We experimented with one naive approach, being average Word2Vec, two general-domain approaches, being Sent2Vec and SentenceBERT, and two domain-specific approaches, being BioSentVec and BioBERT.

The ROUGE scores for the five embedding types are listed in Table 5.1. Both average Word2Vec and BioBERT performed significantly better than the other embeddings for the average ROUGE-1 metric. The differences between the average

ROUGE-2 and ROUGE-L scores were less. Figure 5.1 presents box plots of the ROUGE-1F scores for the 75 summaries that were generated with each embedding type. As we can see, BioBERT has the highest median and maximum value. We selected BiobERT as the embedding approach for our system since it performed slightly better than the others.

Table 5.1: Average ROUGE scores for the sentence embedding approaches. The best ROUGE scores are bolded.

Embedding	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Average Word2Vec	30.77	10.78	26.88
Sent2Vec	29.63	10.53	26.31
BioSentVec	29.77	10.67	26.55
SentenceBERT	29.74	10.26	26.85
BioBERT	**30.79**	**10.89**	**26.98**

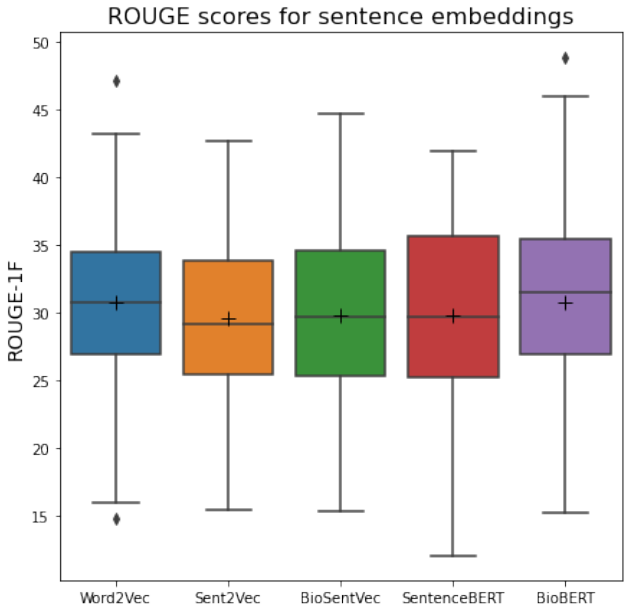


Figure 5.1: Box plot of ROUGE-1F scores for the different sentence embeddings, where the mean is represented in the plots with a +.

5.1.2 Clustering

We tested out K-means, HAC, Spectral Clustering with both cosine and Euclidean as similarity measures for the clustering step. In addition, we tried the pipeline without any clustering of sentences. The results from the different runs are presented in Table 5.2. The Figure 5.2 shows the variability of the ROUGE-1F values for the different clusterings. HAC achieved the best mean ROUGE scores and is therefore

selected as our clustering algorithm. In the box plot, HAC also has an interquartile range that is higher than the other clusterings. Cosine as a measure achieved best on both K-means and HAC. It is also worth mentioning that no clustering achieved similar ROUGE scores to many of the clustering algorithms.

Table 5.2: Average ROUGE scores for the clustering approaches.The best ROUGE scores are bolded.

Model	**Similarity measure**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
K-means	Cosine	30.79	10.89	26.98
K-means	Euclidean	29.63	10.87	26.47
HAC	Cosine	**31.84**	**12.13**	**28.72**
HAC	Euclidean	29.74	10.36	26.16
Spectral clustering	Cosine	29.73	10.65	26.17
Spectral clustering	Euclidean	30.23	10.65	26.87
No clustering	-	29.87	10.89	26.99

5.1.3 Sentence Scoring

After the sentences have been divided into clusters, they are assigned a score. In the ablation study, we experimented with four different scoring methods. The results are presented in Table 5.3. The combination score achieved the highest scores for all three ROUGE metrics with a relatively great margin. From Figure 5.3 one can see that the combination score has more summarizations with very high scores, and the lower whisker is shorter than the other.

Table 5.3: Average ROUGE scores for the sentence scoring approaches. The best ROUGE scores are bolded.

Scoring	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Random	29.45	9.40	26.03
Distance to centroid	30.50	10.98	27.11
LexRank	29.11	10.37	26.34
Combination	**31.84**	**12.13**	**28.72**

5.1.4 Sentence Selection

The final step in the summarization pipeline is to select sentences from the clusters. We tested out three different approaches; local selection, global selection, and top sentences proportional to the cluster size. When selecting proportional to the cluster size, we achieved the highest average ROUGE scores, as presented in Table 5.4. Its median is slightly lower than for global and local selection, shown in Figure 5.4. However, it has higher values for the interquartile range.

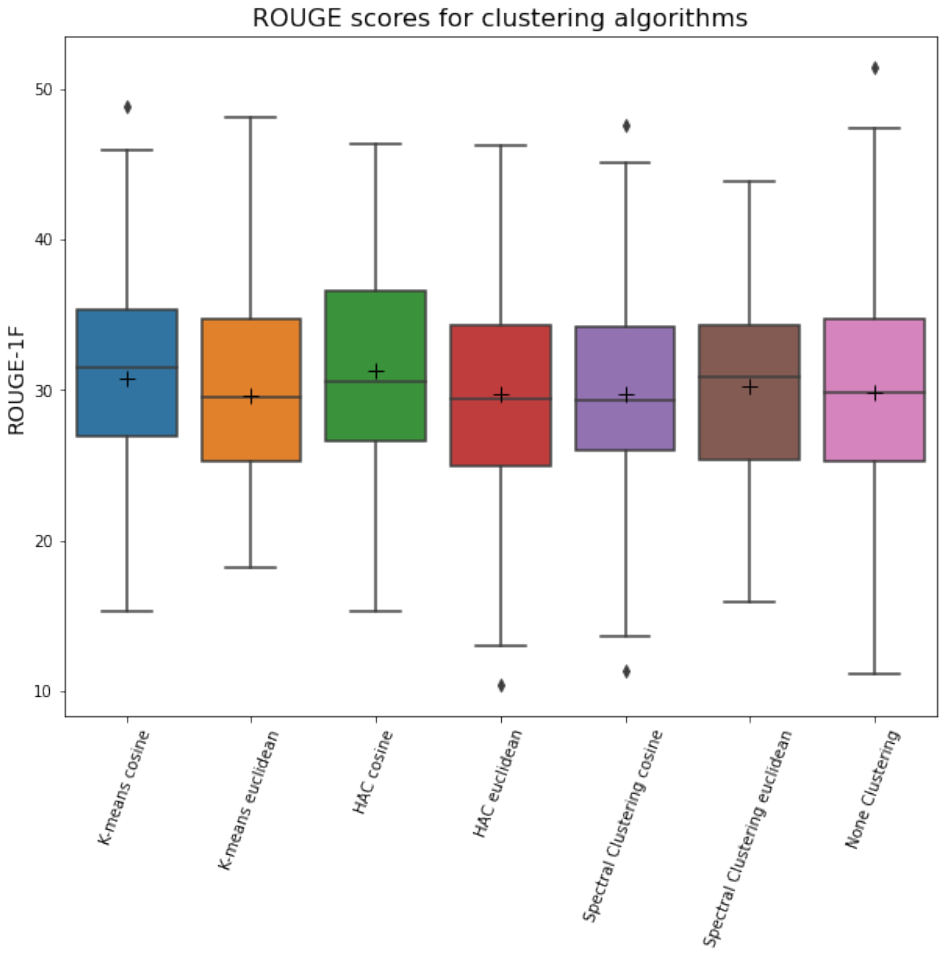


Figure 5.2: Box plot of ROUGE-1F scores of the different clustering algorithms, where the mean is represented in the plots with a +.

Table 5.4: Average ROUGE scores for sentence selection approaches. The best ROUGE scores are bolded.

Selection	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Local selection	30.76	11.37	28.37
Global selection	30.46	11.32	27.91
Proportional	**31.84**	**12.13**	**28.72**

5.1.5 Final Pipeline

The final pipeline decided in the ablation study is illustrated in Figure 5.5. For the sentence embedding step, we chose BioBERT, and for the clustering step, HAC with cosine. For scoring, the combination score was best, and lastly, for selecting sentences, we chose proportional sentence selection.

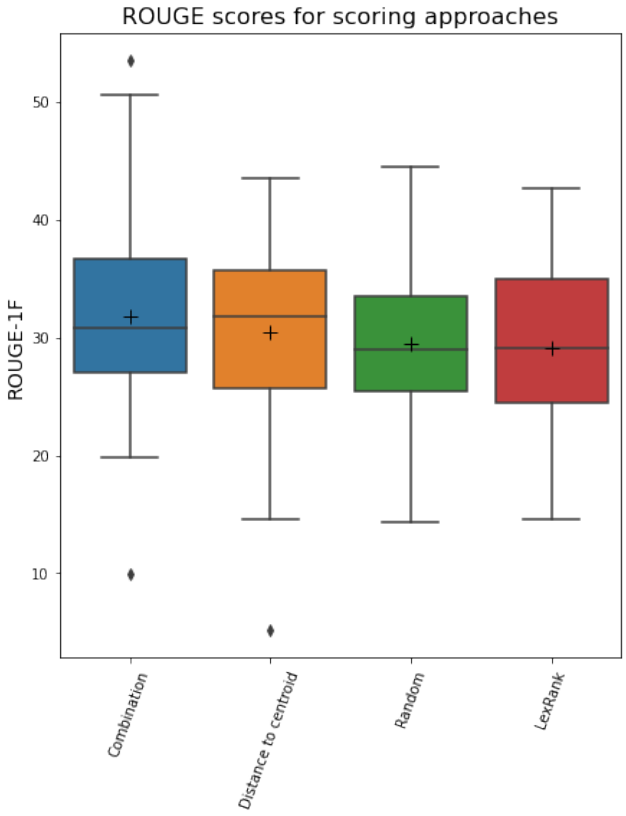


Figure 5.3: Box plot of ROUGE-1F scores for the different sentence scoring approaches, where the mean is represented in the plots with a +.

5.2 Abstractive Step

5.2.1 Pegasus

After selecting the different methods for our summarization pipeline, we observed that Pegasus generated relatively short summaries. The average summary length was 145 words compared to the average gold summary length, which was 364 words. After setting the parameter `length_penalty` on the Pegasus model, the summaries generated got an average length of 171 words. Additionally, the ROUGE-1F score increased by 6.28% with a new score of 34.26. The ROUGE-2F was 12.0, and the ROUGE-LF was 29.18.

We experimented with different numbers of sentences selected for the extractive summary. Figure 5.6 show that more sentences give higher ROUGE scores, but at 40 sentences, the graph levels off, and there is no further improvements in the ROUGE score.

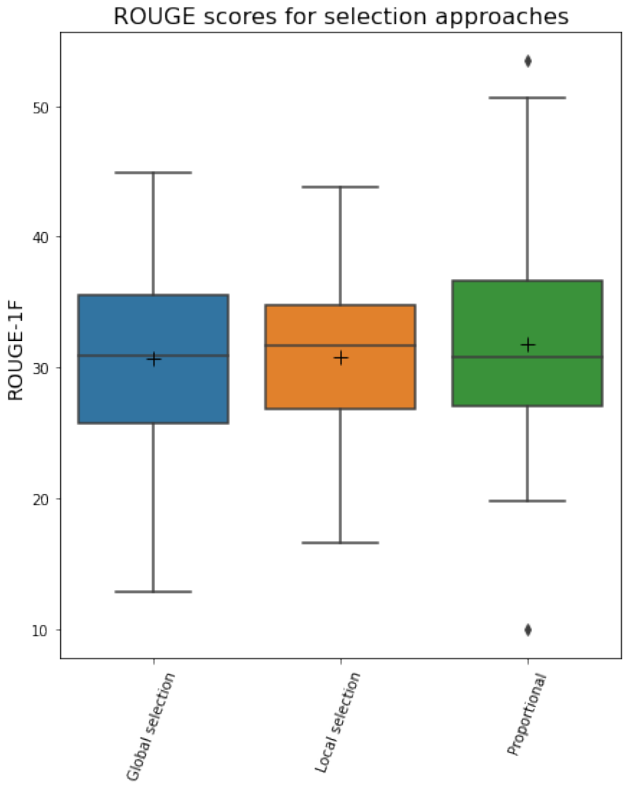


Figure 5.4: Box plot of ROUGE-1F scores for the different sentence selection approaches, where the mean is represented in the plots with a +.

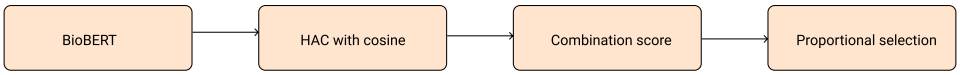


Figure 5.5: Final pipeline for the proposed system decided in the ablation study.

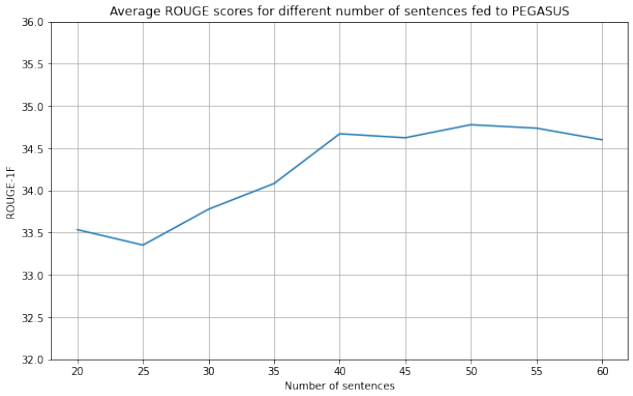


Figure 5.6: Line plot of ROUGE-1F scores with number of sentences fed to Pegasus

Table 5.5: Average ROUGE scores using Pegasus.

Dataset	ROUGE-1	ROUGE-2	ROUGE-L
PubMed	34.26	12.0	29.18
CP	17.53	2.33	11.82

5.2.2 BigBird-Pegasus

After the release of BigBird-Pegasus, we performed several experiments using it to generate abstractive summaries instead of Pegasus. The experiments differed in the number of sentences that were selected for the extractive summary. Figure 5.7 show the average ROUGE-1F scores that were obtained for these experiments. The highest ROUGE-1F score, being 39.75, was achieved when selecting 90 sentences.

As a final evaluation, we ran the pipeline on 1000 PubMed articles. The average ROUGE-1F score of the 500 generated summaries was 38.91. The duration of the run was around 24 hours.

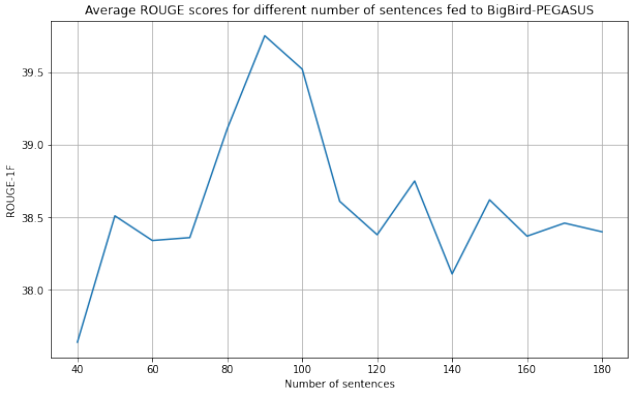


Figure 5.7: Line plot of ROUGE-1F scores with different number of sentences fed to BigBird.

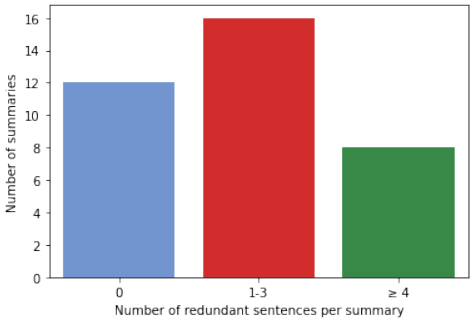
Table 5.6: Average ROUGE scores using Bigbird-Pegasus.

Dataset	ROUGE-1	ROUGE-2	ROUGE-L
PubMed	39.75	16.42	33.10
CP	22.83	3.33	15.06

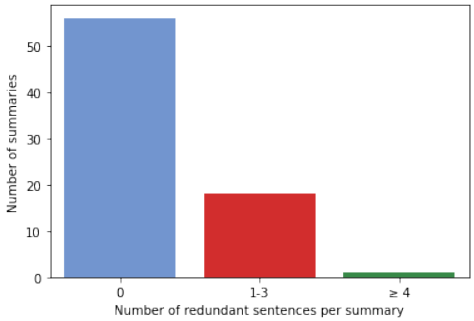
5.3 Redundancy Evaluation

We performed an informal evaluation procedure to detect redundancy in the summaries that were generated by BigBird-Pegasus for both the PubMed dataset and the

CP dataset. The summaries were divided into three categories based on the number of redundant sentences it contained. Figure 5.8a presents the evaluation results for the CP dataset summaries, while Figure 5.8b present the evaluation results for the PubMed dataset summaries. For the CP summaries, 33.3% contains no excessive sentences, 44.4% contains between one and three excessive sentences, and 22.2% contains four or more excessive sentences. For the PubMed summaries, the three categories contain 74.7%, 24.0%, and 1.3% redundant sentences, respectively.



(a) Redundancy in the CP dataset summaries.



(b) Redundancy in the PubMed dataset summaries.

Figure 5.8: Bar charts showing the redundancy in the summaries.

Chapter 6

Discussion

This chapter will first discuss our results from the ablation study, how the abstractive step affects the processing flow, and our validation of the system, including the datasets. Next, as a further evaluation, we study three generated summaries and present our observations. Finally, we answer our research questions presented in Section 1.2.

6.1 Ablation Study

In order to find the best pipeline for our summarization, we performed an ablation study. This narrows down the number of experiment runs needed to determine the best approaches in each step. However, a drawback with the method is that we evaluate every step based on the difference in ROUGE scores obtained from the whole pipeline. Therefore, it is difficult to isolate the different approaches' contributions. Due to the time and number of runs, we chose to input 150 documents into the system. This could be considered as a small selection, but we think it was sufficient for our experiments and necessary due to our time limitations.

The experiments conducted for the different sentence embeddings achieved only slightly different ROUGE scores. The difference may not be sufficient to state that one embedding is better suited for our task.

Contrary to expectations, we did not find a significant difference between the domain-specific embeddings and the general-domain embeddings. In the papers presenting the biomedical embeddings, they achieved better results on the biomedical-specific experiments (Chen et al., 2019; Lee et al., 2020). However, when used by others in the literature on slightly different tasks, it varies whether the domain-specific or general model performs best (Ju et al., 2020; Moradi, Dorffner et al., 2020). This demonstrates that powerful deep learning models trained on general-purpose corpora could just as well be directly applicable to the biomedical domain.

Furthermore, averaging Word2Vec, the most naive embedding method, got sur-

prisingly good results. A possible explanation can be that Word2Vec embeddings have a dimension of 100 while the other embeddings have 700, as it may be challenging to cluster very high-dimensional data (Assent, 2012).

In Figure 6.1 we have reduced the number of dimensions to two dimensions using principal component analysis (PCA) to illustrate the clusterings for the first summarization for different sentence embeddings. We can observe that different embeddings result in different clusterings. The figure illustrates that the choice of one step can affect the next step in the pipeline, which can also be the case in other parts of the processing flow.

As stated in Assent (2012) it follows from "the curse of dimensionality" that distance and similarity measures lose their discriminative power. Moreover, since distance measure plays a vital role in clustering, it may become difficult to group the high-dimensional data. The high-dimensional data can explain our challenges of determining K. We got an average silhouette score of 0.1, that also indicates that it is challenging to cluster the data. A score near zero means that the data point might as well belong to another cluster (Shahapure & Nicholas, 2020).

Further, a limitation of the clustering step is that the experiments to determine K were only conducted with K-means. However, it would have been very time-consuming to determine K for every clustering algorithm. A further improvement to the pipeline could be to dynamically determine K to match the number of topics in the documents.

HAC with cosine as a measure achieved relatively better scores than the other clustering experiments. One difference between HAC with cosine and the other implementations is how distance to clusters is computed. All clustering implementations are based on distances to the centroid except the HAC with cosine implementation, which uses complete linkage where the distance between the farthest points is used.

The scoring step decides what sentences are important in the clusters. The combination score got the best ROUGE score. Interestingly, the only difference between the combination score and distance to the centroid is the novelty score parameter in the combination score. The results indicate that the novelty parameter contributes to a 4.4% better ROUGE-1F score. Further, it is surprising that random scoring achieves relatively high scores, even better than LexRank scoring. The original LexRank method considers all sentences in the documents together while we compute the scores cluster by cluster. One reason why LexRank scoring is underperforming can be that it does not reach its full potential when the scoring is computed inside clusters.

A reason for random's good results can be that regardless of scoring, the Pegasus model is fed with 40 sentences. Additionally, the big clusters are emphasized with proportional selection, and Pegasus can capture the important topics, regardless of the random scoring of sentences.

For the selection step, proportional got significantly better ROUGE scores than the others. The proportional selection emphasizes the bigger clusters and supports the assumption that large clusters contain more important topics and that small clusters should contribute with some sentences to represent all topics. The local top k suffers from the limitation that if the cluster contains less than k sentences, the extractive summary will decrease with the corresponding number of sentences. This may result in a short extractive summary, and Pegasus will not be fed maximum input length, which we found from Figure 5.6 was optimal. Global selection selects the highest scored sentences in the two documents. However, a sentence is scored in comparison to the other sentences inside its cluster. It can be that the scores are not comparable across clusters, and that global selection therefore achieves the lowest ROUGE score of the three selection methods.

In general, the box plots presented in Section 5.1 show that the summarizations differ much in the ROUGE score. The reason for this can be that documents can vary in degree of complexity. Optimally we want a more reliable system where it is possible to generate good summaries for documents of all lengths. Also, the naive methods performed relatively well in our experiments. One naive step might not make much of an impact on the entire system's ability to make summaries. Especially, the pre-trained model at the end of the processing flow is powerful and can manage to generate summaries even with a preceding naive step.

6.2 Abstractive Step

As described in Section 4.3.6, we used two different pre-trained language models for the generation of the abstractive summaries; Pegasus and BigBird-Pegasus. We used Pegasus in the abstractive step of the summarization system. As presented in Section 5.2.1, Pegasus' results are improved when we add the `length_penalty` parameter. This improvement corresponds to what Figure 6.2 presents. The figure shows a positive correlation between the length of the generated summary and the ROUGE score.

As BigBird-Pegasus was released on HuggingFace at the beginning of May 2021, it was not used for the experiments in our ablation study. However, as the results from Section 5.2 show, BigBird-Pegasus achieved significantly better ROUGE scores than Pegasus for both datasets. A reason for this substantial improvement can be explained by the amount of input text that BigBird-Pegasus can handle compared to Pegasus. The main difference between BigBird-Pegasus and Pegasus lies in the attention mechanism in the encoder, as explained in Section 4.3.6. This difference results in a larger input capacity for BigBird-Pegasus. As the average length of the generated summaries is the same for both language models, it can be assumed that using an increased amount of sentences as input to the model results in higher ROGUE scores. This assumption is also supported by the results shown in Figure 5.7, where we can see that Pegasus' results increased as we increased the

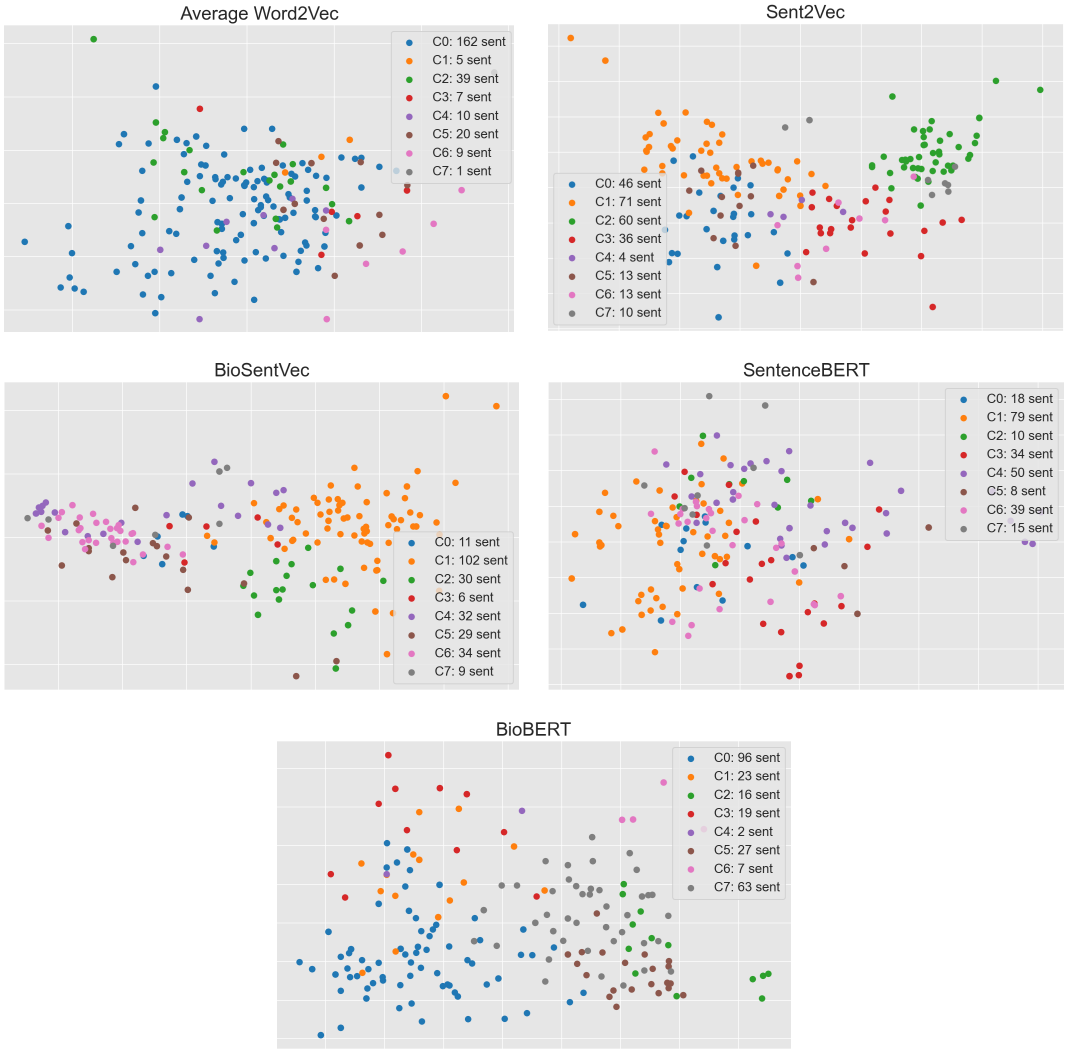


Figure 6.1: Visualization of the HAC clustering from the first summarization. The embeddings are decomposed to two dimensions using PCA.

number of sentences selected for the extractive summary. However, for BigBird-Pegasus, this is only somewhat true. The ROUGE-1F scores for BigBird-Pegasus largely increased from selecting 40 sentences to selecting 90 sentences for the extractive summary. After that, the scores started to decrease. A reason for this reduction can be that the lengthy extractive summaries contain more irrelevant sentences than the shorter ones. It may also indicate that the extractive step in our summarization pipeline has a positive contribution to the abstractive step.

Even though the pre-trained models are a good contribution to our summarization system, they have some obvious drawbacks. The first being that these models can be seen as black boxes. As explained in Section 2.4, the lack of transparency and interpretability is unfortunate when creating systems that can be used for supporting decision-making. Another major drawback is the possibility of hallucinations

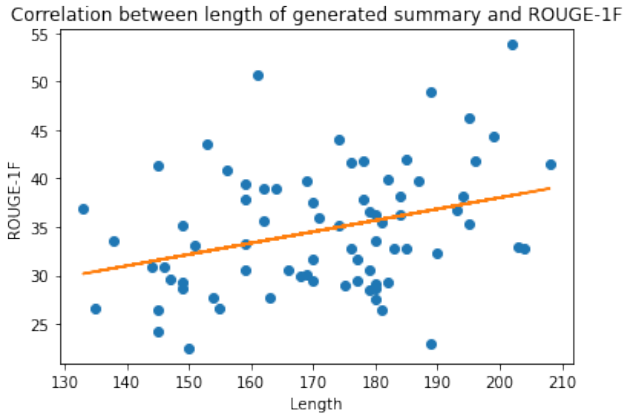


Figure 6.2: Correlation between generated summary lengths and ROUGE scores is 0.326457.

by the models. The pre-trained models might create summaries that include untrue information, which can be a critical source of error in a decision support system.

6.3 Validation

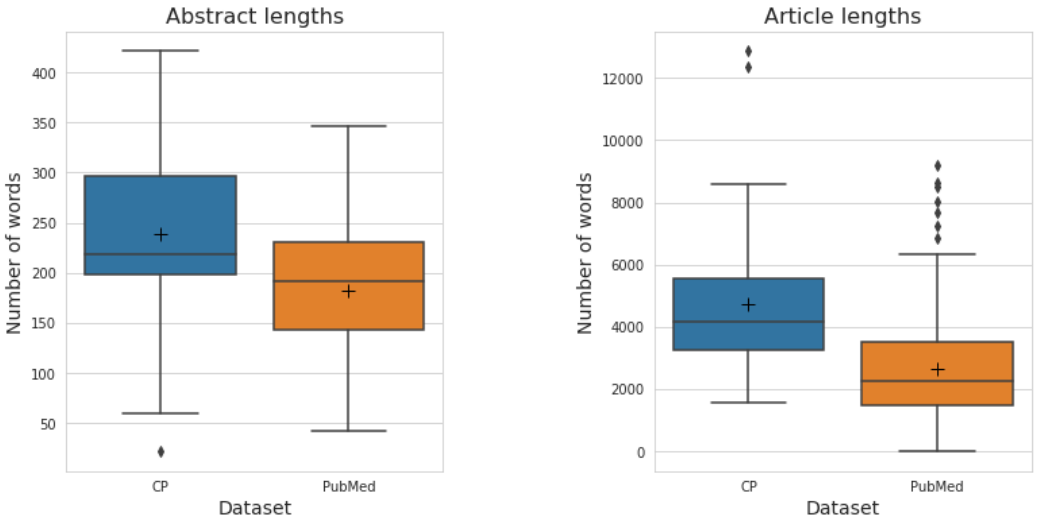
Validation of the system is an essential step in order to understand our system's performance and other findings. This section will discuss the datasets, the ROUGE metric, the redundancy evaluation we conducted, and three example summaries of different quality.

6.3.1 Dataset

Evaluating the performance of a biomedical multi-document summarization system is not easy, as there does not exist a dataset adapted to this task. Several datasets exist for evaluating multi-document summarization, but none relevant as most of them focus on shorter, general-domain articles, and we want our system to handle long, biomedical articles. The main disadvantage of lacking a suitable dataset is that we have no optimal gold summary to compare with our generated summaries. With no such dataset, it is not easy to evaluate how adjustments affect our system and how it performs against other similar systems, which is a considerable drawback. Our solution with concatenating the abstracts and use them as gold summaries is not optimal, but it gives us an indication of our system's performance.

As presented in Section 4.4, we evaluate our system using two different datasets; one containing articles from PubMed and one containing CP-related articles. A comparison of the CP and PubMed datasets is presented in Figure 6.3. Figure 6.3a shows the length of the abstracts for the two datasets, while Figure 6.3b shows the

length of the articles. The CP dataset contains longer abstracts and articles than the PubMed dataset, especially for the articles that are on average 77% longer. The two box plots also show that both datasets have a large variability. In addition, we can see from Figure 6.3b that there are large outliers for the article lengths in the CP and the PubMed dataset.



(a) Length of abstracts. The median length of an abstract from the CP dataset is 218 words, while from the PubMed dataset is 192 words. The mean of the abstract length from the CP dataset is 239 words and from the PubMed dataset is 182 words.

(b) Length of articles. The median length of an article from the CP dataset is 4143 words, while from the PubMed dataset is 2252 words. The mean of the article length from the CP dataset is 4719 words and from the PubMed dataset is 2656 words.

Figure 6.3: Box plots showing the average lengths of abstracts and articles in the CP and PubMed datasets. The mean is represented in the plots with a +.

Another essential difference between the CP dataset and the PubMed dataset is that all documents in the CP dataset are related to the same topic. In contrast, the documents in the PubMed dataset vary in topic. As a result, the gold summaries in the CP dataset consist of two abstracts related to the same topic. For the PubMed dataset, a gold summary can consist of two concatenated abstracts about two completely different topics.

As the results from Section 5.2 show, the ROUGE values obtained for both Pegasus and BigBird-Pegasus using the CP dataset are much lower than when using the PubMed dataset. The ROUGE-1F score for Pegasus is reduced by 48,8%. For BigBird-Pegasus, it is reduced by 42,6%. This reduction in performance can be explained by the substantial difference in the length of the gold summaries and articles between the two datasets, as Figure 6.3 shows. Figure 6.4 shows that there is a negative correlation between the length of an article and the ROUGE-1F score obtained, i.e., longer articles are more challenging to summarize. Another important aspect is that the gold summaries are longer than the summaries generated by

Pegasus and BigBird-Pegasus. A gold summary is on average $239 * 2 = 478$ words for the CP dataset and $182 * 2 = 364$ words for the PubMed dataset. However, both models generated summaries that contained 170 words on average. Generating summaries that are less than half the length of the gold summaries negatively affects the F-measure of the ROUGE scores, as the recall value will be decreased. This is also presented in Figure 6.5, which shows that lower ROUGE-1F scores are obtained when the gold summary is longer.

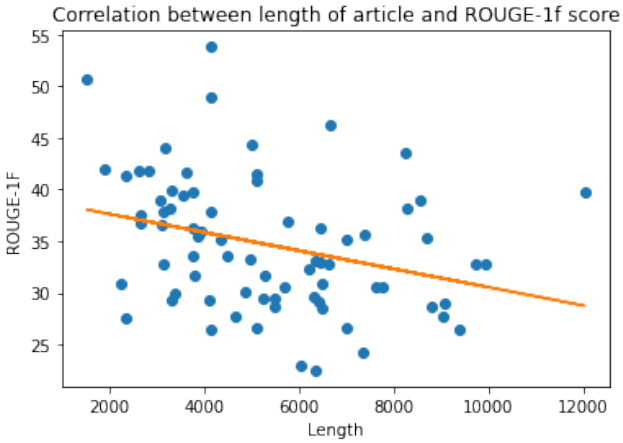


Figure 6.4: Correlation between article lengths and ROUGE scores is -0.309629.

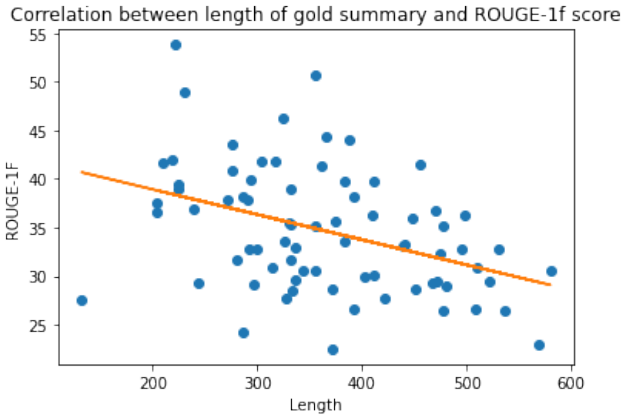


Figure 6.5: Correlation between gold summary lengths and ROUGE scores is -0.40006.

6.3.2 Evaluation

ROUGE

As described in Section 4.4, we used the ROUGE-metric to evaluate our experiments. The ROUGE-metric is one of the most used evaluation metrics for automatic

text summarization. However, it has some obvious drawbacks. The ROUGE-metric requires a gold standard summary to compare with the generated summary. Having human-written gold summaries to compare with would be optimal, but this is an expensive process concerning time and requires domain knowledge. It should also be noted that the definition of a good summary is a subjective opinion. People can have very different views on what is the most important information in a document, and therefore, different preferences of what a gold summary should contain. Hence, evaluating summaries is a challenging task.

Another drawback with the ROUGE-metric is that it only measures word overlap. Therefore, it is possible to achieve high ROUGE scores for a poorly written summary. It is not able to capture how well-written or fluent a summary is. Another aspect is that we create abstractive summaries in our experiments. This means that the summaries can contain other words or phrases than initially used in the document, which might affect the results as the ROUGE measure looks at identical words.

Considering that we only use the ROUGE metric to evaluate our summaries, we do not receive any indication of how much a document contributes to a generated summary. Since we try to summarize two documents, we want the generated summary to contain information from both documents. Evaluating each document's contribution is challenging, especially considering that we create abstractive summaries from biomedical documents.

Redundancy Evaluation

When creating summaries, it is desirable that they are concise, consistent, and only contain the most important information from the documents. As a result of this, the summary should contain as little redundancy as possible. Measuring redundancy is challenging, and there is no ideal solution for doing it. Human evaluation using medical experts would be optimal for estimating the amount of redundancy in our summaries, but this is very time-consuming.

In Section 4.4.2, we described how we perform an informal redundancy evaluation ourselves. Section 5.3 presents the results from the redundancy evaluation. As we can see, there is a considerable difference between the two datasets. The summaries generated for the PubMed dataset contained much less redundancy than the summaries for the CP dataset. 33.3% of the CP summaries were non-redundant, while the number for the PubMed summaries was almost 75%. For the summaries containing between 1-3 redundant sentences and the summaries containing four or more redundant sentences, the difference between the datasets was around 20%. It should be noted that the PubMed dataset generated twice as many summaries as the CP dataset.

The language models are black boxes, and it is difficult to understand why they generated more redundant sentences for one of the datasets compared to the other. A

possible reason for the redundancy in the CP summaries can be that the model summarizes two documents related to the same topic. For the PubMed dataset, however, the model summarizes two documents about different topics. The extractive summaries for the PubMed dataset might contain more diverse sentences, while for the CP dataset, the sentences are more alike, which can result in excessive sentences. When we evaluated redundancy in the summaries, we noticed that the language model generated the same sentence multiple times, with minimal variations. Even though the generative language models have made great progress in last couple of years, they can still suffer from problems such as adding duplicate sentences to a summary.

There are some significant limitations with our redundancy evaluation that can affect the evaluation results. One of the major limitations is regarding bias, as we performed the evaluation ourselves and did not have other people evaluating the summaries. Further, it is a drawback that we only checked for more or less identical phrases and sentences. We did not look at the overall content of the sentences, as we do not have a medical background. This lack of understanding medical texts results in the possibility that there were more redundant sentences in the summaries that we could not detect.

6.3.3 Generated Summaries

The summarization pipeline produced summaries of varying quality. We selected three generated summaries of different quality from the CP dataset to further understand how the system performs.

Example 6.1 is, according to ROUGE-1F, the best summary in the CP dataset with a score of 35.02. However, the summary contains moderate redundancy. The redundant sentences are highlighted with italics in Example 6.1. In addition to redundancy, two of the sentences are contradictory. The first sentence states, "there were no differences in mri findings..." while the next states, "however, there were differences in mri findings...". The sentences show that the system can generate total opposite facts. Hallucinating abstractive summarization models is a known problem in the literature (Mao et al., 2020). The untrue facts can hinder the applications from being trusted in a real-world application, which is a considerable drawback. The summary also suffers from an incomplete sentence at the end of the summary.

Further, Example 6.2 is the highest scored non-redundant summary. We consider this as a coherent summary that fulfills the criteria of a good quality summary. This example shows the potential of our system. On the other hand, Example 6.3, shows the opposite, which has obtained a ROUGE-1F score of 14.83. Furthermore, with five excessive sentences, the summary is considered highly redundant. Additionally, the summary contains repeating words considered trash at the end of the summary. Example 6.3 clearly shows that the system may also generate non-successful summaries.

In Appendix A the examples' corresponding gold summaries are attached. As mentioned in Section 6.3.2, ROUGE does not capture the contribution from the different articles to the generated summary. From what we can observe for these three summaries, when comparing gold summaries with the selected generated summaries, the generated summaries contain information mainly from one article. This contradicts with the goal of multi-document summarization, which should cover the information about all topics in the documents.

Overall, it seems that BigBird experiences bigger problems generating good quality sentences at the end of the summaries. Uncomplete sentences, redundant sentences, and trash words tend to appear at the end.

Example 6.1: High ROUGE-1F (35.02) and moderate redundant

we investigated the relationship between pre- and postnatal brain magnetic resonance imaging (mri) findings and motor development in a cohort of premature infants with cerebral palsy . this was a prospective cohort study in a tertiary care children 's hospital . mri was obtained at 36 weeks of gestation for all premature infants with cerebral palsy . the motor development of each child was assessed by measuring the amount of movement and postural control of the fingers and toes using a split - field motor cycle test . children were grouped according to age at mri : 6 months , 6 to 12 months , and > 12 months of age . postural control was assessed by measuring the amount of movement of the index finger and toes using a split - field motor cycle test *. there were no differences in mri findings between premature infants with and without cerebral palsy at age 6 months and 12 months . however , there were differences in mri findings between premature infants with and without cerebral palsy at age 6 months and 12 months of age . there were also differences in mri findings between premature infants with cerebral palsy and those without cerebral palsy at age 6 months and 12 months of age* . in conclusion , this study

Example 6.2: Highest ROUGE-1F-scored (23.19) non-redundant summary

objectivethe aim of this study was to evaluate the safety of online collaborative learning for infants with autism spectrum disorder (asd) in india.methodsa total of 78 infants with asd and their parents were enrolled in the study . parents completed a self - report questionnaire that included items on demographics , social support , and asd . the infants were assessed using a collaborative learning platform called platform for active learning (pkool) . the pkool consists of three parts . the first part is a selfdescription of the child s behavior during the first six months of life , followed by a description of behavior during the last six months . the second part of the study focused

on the safety of online collaborative learning for infants with asd.resultsthe overall safety of the platform was good . there were no safety issues related to the platform . most of the children had a good description of their behavior during the first six months of life . however , after the sixth month , the safety of the platform declined.conclusionthe findings of this study suggest that online collaborative learning for infants with asd can be safe . further studies are needed to evaluate the safety of the

Example 6.3: Low ROUGE-1F (14.83) and highly redundant

infantile spasms are a common cause of developmental delay and disability in infants . the infantile spasms are characterized by loss of control of body movements , resulting in death of the infant . there are a number of risk factors for the development of infantile spasms , including gestational age , low birth weight , infectious and metabolic causes , as well as genetic predisposition . the mechanisms underlying the infantile spasms have not been fully elucidated . this is a critical review of our current understanding of the mechanisms underlying infantile spasms . *the first part of the review covers the development of our current understanding of the mechanisms underlying infantile spasms . the second part of the review covers the development of our current understanding of the mechanisms underlying the infantile spasms . the first part of the review covers the development of our current understanding of the mechanisms underlying the infantile spasms . the second part of the review covers the development of our current understanding of the mechanisms underlying the infantile spasms . the first part of the review covers the development of our current understanding of the mechanisms underlying the infantile spasms . the second part of the review covers the development of our current understanding of the mechanisms underlying the infantile spasms.imagesfigure 1figure 2figure 3figure 4*

6.4 Answering Research Questions

In this section, we will answer the research questions defined in Section 1.2. As stated in our main research question, this project was conducted to evaluate how multiple biomedical documents could be automatically summarized and which methods were most suitable for this task. The research questions will be answered in light of the results presented in Chapter 5.

RQ. How to generate multi-document summarization from biomedical texts using text summarization and text mining techniques?

We have based this project on the fact that automatic multi-document summarization of biomedical articles is a challenging task and that there exists no suitable solution for it. Current state-of-the-art systems focus on either single-document summarization of biomedical articles or multi-document summarization of shorter, general-domain texts such as news articles. In order to solve the problem, we created a hybrid summarization system that combines several methods within text mining and machine learning to create abstractive summaries. Our system achieved a ROUGE-1F score of 39.75, showing great potential for supporting decision-making within the biomedical domain and validating predictions from machine learning models.

RQ1. How can sentence embeddings capture semantics from biomedical texts?

We utilized neural sentence embeddings for the representation of the input documents. We experimented with general-domain and domain-specific embeddings to see how sentence semantics could best be captured. The results from the experiments indicated that the type of data the embeddings are trained on has little impact on the system's performance. All five sentence embeddings achieved relatively similar ROUGE-1F scores, with average Word2Vec and BioBERT scoring the highest. The ROUGE scores obtained in the ablation study can indicate that sentence embeddings are a suitable option for representing biomedical text. However, they might not be ideal when combined with clustering due to a large number of dimensions.

RQ2. How can clustering, sentence scoring, and sentence selection improve the process of extracting salient information?

In order to use pre-trained language models for generating abstractive multi-document summaries, we first created extractive summaries of the input documents. The extractive summaries were generated by first clustering the sentence embeddings and then selecting sentences based on a given score. Through the results from our ablation study, we were able to detect which methods within clustering, scoring, and selection were optimal for our process flow. By using HAC with cosine similarity, combination score, and proportional sentence selection, we were able to extract sentences containing more prominent information from the documents than when using the other methods. The clustering step intends to group sentences of similar topics. The scoring step ranks the sentences based on their importance in the document and penalizes similar sentences. Lastly, the selection approach helps select more sentences related to the main topics in the documents while also covering the small topics. Through these techniques, we were able to improve the ROUGE scores for our system.

RQ3. What evaluation methods can be used to verify that the summaries are

non-redundant and preserve the most important information?

Evaluating abstractive summaries regarding how much redundancy they contain and if the most essential information is maintained is difficult. As there is no optimal approach to verify that the generated summaries are non-redundant, we chose to perform an informal evaluation ourselves. Since we do not have a medical background, we could only detect redundant sentences based on the number of identical words to other sentences. Through this evaluation, we detected that some summaries contained little to no excessive sentences, while others contained quite a lot. However, the redundant sentences we detected came from the pre-trained language model generating the same sentence multiple times and not from redundant sentences in the extractive summary.

We used the ROUGE metric to evaluate the amount of important information that the summaries contained. It is not an optimal metric, but it is the most used within text summarization and gave us a good indication of how well our summarization system performed. Using the abstracts as gold summaries enabled us to verify that we included in our generated summaries the information that the author thinks is essential.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have described how automatic summarization of articles can support clinicians in decision-making regarding diagnosticating diseases. We have shown that there does not exist a sufficient or suitable system to solve this. We address the problem by presenting a novel hybrid multi-document summarizer that utilizes different subfields within text mining and machine learning to handle large amounts of input data.

We conducted an ablation study in order to find the optimal methods to include in our processing flow. We have explored how sentence embeddings and clustering, scoring, and selection of sentences can be applied in the extractive part of the system. For the abstractive part, we experimented with two different pre-trained language models. Given the different steps that the processing flow consists of, we conclude with BioBERT embeddings, HAC with cosine similarity, combination score, proportional sentence selection, and BigBird-Pegasus as the best options. Through the different methods we selected, we were able to increase the obtained ROUGE scores for our system.

The experiments that we carried out resulted in several interesting findings. For the usage of sentence embeddings, we conclude that using domain-specific embeddings is not necessarily a better option than general-domain embeddings for our system. We have also shown that using an increased amount of sentences as input to the language model can result in higher ROGUE scores and possibly better summaries. However, there exists an optimal number of sentences that should be passed to the pre-trained models. With this in mind, we conclude that the extractive step in our process flow has a positive contribution to the overall system. In addition, it is an essential step to enable the summarization of multiple biomedical articles.

The experiments conducted have shown that our extractive pipeline works properly but that the pre-trained language model is not fine-tuned well enough for

our task. We have detected some limitations regarding the use of pre-trained language models, as our problems related to redundancy and hallucinations occur in the abstractive step.

We recognize that our system has some limitations. Nevertheless, the processing flow presented in this thesis contributes to building a solid foundation for creating well-written summaries of multiple biomedical articles. The system is scalable and has the ability to process several documents at a time. The system is also built in such a way that it is generic and does not require a specific structure for the documents. It has the potential to help explain and validate the predictions from machine learning models, thereby supporting decision-making within the biomedical domain.

7.2 Future Work

As a result of the work conducted in this thesis and based on our previous discussion, we propose that further research should be undertaken in the following areas.

Dataset As stated earlier, one of the most notable drawbacks of our research is the lack of a dataset for biomedical multi-document summarization. Creating a multi-document summarization dataset containing biomedical articles is a demanding task with respect to time and human resources. However, we believe that the creation of such a dataset is vital for further research. It will enable us to evaluate our system more correctly and, just as importantly, compare it to other systems. It will also make it simpler to see how different changes can improve the system. Finally, the creation of a biomedical MDS dataset might facilitate different numbers of documents to be used as input. In that way, we can evaluate the system's performance when used on more than two documents, which is necessary for future research.

Pre-trained model An important issue to resolve for future studies is the limitations we experienced with the pre-trained language models. As mentioned previously, the pre-trained models might hallucinate and produce sentences containing false information. We also experienced that the models generate the same sentences multiple times in one summary and that the summaries contain information from one document mainly. The development of a biomedical MDS dataset can help minimize these issues. It can benefit future research by making it possible to fine-tune the language model on a more suitable dataset. We are optimistic that fine-tuning can help reduce the excessive sentences and hallucination of information generated in the abstractive summaries and ensure that information from all documents is included in the summaries.

Sentence representation and clustering From the experiments we conducted in our ablation study, we discovered that the interaction between the sentence embeddings and the clustering was not optimal and that there is room for improvements. Therefore, we believe that further experimental investigations are needed in order to enhance this part of the system. Further research can be done on testing out other types of sentence representations or use dimensionality reduction techniques to reduce the number of dimensions for the embeddings.

Bibliography

Adadi, A. & Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, *6*, 52138–52160.

Adde, L. (2019). Fra klinisk observasjon til kunstig intelligens: Fremtidensoppfolging av sykenyfodte?

Adde, L., Helbostad, J. L., JENSENIUS, A. R., Taraldsen, G., Grunewaldt, K. H. & Støen, R. (2010). Early prediction of cerebral palsy by computer-based video analysis of general movements: a feasibility study. *Developmental Medicine & Child Neurology*, *52*(8), 773–778.

Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A. & Idris, N. (2019). Cosum: Text summarization based on clustering and optimization. *Expert Systems*, *36*(1), e12340.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. & Kochut, K. (2017a). Text Summarization Techniques: A Brief Survey. *arXiv*.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. & Kochut, K. (2017b). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.

Assent, I. (2012). Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(4), 340–350.

Bahdanau, D., Cho, K. & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Bai, S. & An, S. (2018). A survey on automatic image caption generation. *Neurocomputing*, *311*, 291–304.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R. & Herrera, F. (2020). Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115.

Bengio, Y., Ducharme, R., Vincent, P. & Janvin, C. (2003). A neural probabilistic language model. *The journal of machine learning research*, *3*, 1137–1155.

Bowman, S. R., Angeli, G., Potts, C. & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y.-H., Strope, B. & Kurzweil, R. (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Chen, Q., Peng, Y. & Lu, Z. (2019). Biosentvec: Creating sentence embeddings for biomedical texts. *2019 IEEE International Conference on Healthcare Informatics (ICHI)*.

Chiu, B., Crichton, G., Korhonen, A. & Pyysalo, S. (2016). How to train good word embeddings for biomedical nlp. *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, 166–174.

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. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734.

Clark, K., Luong, M.-T., Le, Q. V. & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

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. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *2*, 615–621.

Cohen, P. R. & Howe, A. E. (1988). How evaluation guides ai research: The message still counts more than the medium. *AI Magazine*, *9*(4), 35.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L. & Bordes, A. (2018). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Dan, S., Xu, Y., Yu, T., Siddique, F. B., Barezi, E. & Fung, P. (2020). Caire-covid: A question answering and query-focused multi-document summarization system for covid-19 scholarly information management.

Das, D. & Martins, A. F. T. (2007). *A Survey on Automatic Text Summarization* (tech. rep.). Carnegie Mellon University.

Deng, L. & Liu, Y. (2018). *Deep Learning in Natural Language Processing* (1st). Springer Publishing Company, Incorporated.

Deshpande, M. (2020). Language Modeling — I. Next word prediction using language... | by Mandar Deshpande | Towards Data Science.

Devlin, J. & Chang, M.-W. (2018). Google AI Blog: Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing.

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*.

Dong, Y., Su, H., Zhu, J. & Zhang, B. (2017). Improving Interpretability of Deep Neural Networks With Semantic Information. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dosilovic, F. K., Brcic, M. & Hlupic, N. (2018). Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 210–215.

Eisenstein, J. (2018). Natural language processing.

El-Kassas, W. S., Salama, C. R., Rafea, A. A. & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, *165*, 113679.

Erkan, G. & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, *22*, 457–479.

Ethayarajh, K. (2018). Unsupervised random walk sentence embeddings: A strong but simple baseline. *Proceedings of The Third Workshop on Representation Learning for NLP*, 91–100.

Gidiotis, A. & Tsoumakas, G. (2020). A divide-and-conquer approach to the summarization of academic articles. *arXiv preprint arXiv:2004.06190*.

Goldberg, X. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *6*, 1–116.

Goodman, B. & Flaxman, S. (2017). European union regulations on algorithmic decision making and a "right to explanation". *AI Magazine*, *38*(3), 50–57.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J. & Poon, H. (2020). Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing.

Haider, M. M., Hossin, M. A., Mahi, H. R. & Arif, H. (2020). Automatic text summarization using gensim word2vec and k-means clustering algorithm. *2020 IEEE Region 10 Symposium (TENSYMP)*, 283–286.

Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B. & Darrell, T. (2016). Generating Visual Explanations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9908 LNCS*, 3–19.

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

Hou, L., Hu, P. & Bei, C. (2018). Abstractive document summarization via neural model with joint attention. In X. Huang, J. Jiang, D. Zhao, Y. Feng & Y. Hong (Eds.), *Natural language processing and chinese computing* (pp. 329–338). Springer International Publishing.

Hu, D. (2020). An introductory survey on attention mechanisms in NLP problems. *Advances in Intelligent Systems and Computing*, *1038*, 432–448.

James, D. (2018). *Introduction to Machine Learning with Python: A Guide for Beginners in Data Science* (1st). CreateSpace Independent Publishing Platform.

Jiang, Z., Srivastava, M., Krishna, S., Akodes, D. & Schwartz, R. (2020). Combining word embeddings and n-grams for unsupervised document summarization. *arXiv preprint arXiv:2004.14119*.

Joshi, A., Fidalgo, E., Alegre, E. & Fernández-Robles, L. (2019). Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, *129*, 200–215.

Ju, J., Liu, M., Gao, L. & Pan, S. (2020). Scisummpip: An unsupervised scientific paper summarization pipeline. *arXiv preprint arXiv:2010.09190*.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R. & Fidler, S. (2015). Skip-thought vectors. *arXiv preprint arXiv:1506.06726*.

Kodinariya, T. M. & Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering. *International Journal*, *1*(6), 90–95.

Lamsiyah, S., El Mahdaouy, A., Espinasse, B. & Ouatik, S. E. A. (2020). An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Systems with Applications*, *167*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H. & Kang, J. (2020). Biobert: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240.

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*.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y. & Li, L. (2020). On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Li, W., Xiao, X., Liu, J., Wu, H., Wang, H. & Du, J. (2020). Leveraging Graph to Improve Abstractive Multi-Document Summarization, 6232–6243.

Lin, C.-Y. (2004). *ROUGE: A Package for Automatic Evaluation of Summaries* (tech. rep.).

Liu, F. & Liu, Y. (2009). From extractive to abstractive meeting summaries: Can it be done by sentence compression? *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 261–264.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L. & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ma, C., Zhang, W. E., Guo, M., Wang, H. & Sheng, Q. Z. (2020). Multi-document summarization via deep learning techniques: A survey. *arXiv preprint arXiv:2011.04843*.

Mao, Y., Ren, X., Ji, H. & Han, J. (2020). Constrained abstractive summarization: Preserving factual consistency with constrained generation. *arXiv preprint arXiv:2010.12723*.

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.

Michie, D. (1968). MEMO FUNCTIONS AND MACHINE LEARNING. *Nature*, *218*(5136), 19–22.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013). Efficient estimation of word representations in vector space. *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.

Miller, D. (2019). Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*.

Moradi, M. (2018). Cibs: A biomedical text summarizer using topic-based sentence clustering. *Journal of biomedical informatics*, *88*, 53–61.

Moradi, M., Dashti, M. & Samwald, M. (2020). Summarization of biomedical articles using domain-specific word embeddings and graph ranking. *Journal of Biomedical Informatics*, *107*, 103452.

Moradi, M., Dorffner, G. & Samwald, M. (2020). Deep contextualized embeddings for quantifying the informative content in biomedical text summarization. *Computer methods and programs in biomedicine*, *184*, 105117.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R. & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1–21.

Nelson, S. J., Powell, T. & Huhmpreys, B. (2001). The unified medical language system (umls) project.

Nilsson, N. J. (1980). *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

Novak, I., Morgan, C., Adde, L., Blackman, J., Boyd, R. N., Brunstrom-Hernandez, J., Cioni, G., Damiano, D., Darrah, J., Eliasson, A.-C. et al. (2017). Early, accurate diagnosis and early intervention in cerebral palsy: Advances in diagnosis and treatment. *JAMA pediatrics*, *171*(9), 897–907.

Page, L., Brin, S., Motwani, R. & Winograd, T. (1999). *The pagerank citation ranking: Bringing order to the web.* (tech. rep.). Stanford InfoLab.

Pagliardini, M., Gupta, P. & Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computa-*

tional Linguistics: Human Language Technologies, Volume 1 (Long Papers), 528–540.

Papanastasopoulos, Z., Samala, R. K., Chan, H.-P., Hadjiiski, L., Paramagul, C., Helvie, M. A. & Neal, C. H. (2020). Explainable AI for medical imaging: deep-learning CNN ensemble for classification of estrogen receptor status from breast MRI. In H. K. Hahn & M. A. Mazurowski (Eds.), *Medical imaging 2020: Computer-aided diagnosis* (p. 52). SPIE-Intl Soc Optical Eng.

Pennington, J., Socher, R. & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Pinto, A., Gonçalo Oliveira, H. & Oliveira Alves, A. (2016). Comparing the performance of different nlp toolkits in formal and social media text. *5th Symposium on Languages, Applications and Technologies (SLATE'16)*.

Plaza, L. & Albornoz, J. (2012). Evaluating the use of different positional strategies for sentence selection in biomedical literature summarization. *BMC Bioinformatics*, *14*, 71–71.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T. & Ananiadou, S. (2013). Distributional semantics resources for biomedical text processing. *Proceedings of Languages in Biology and Medicine*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N. & Huang, X. (2020). Pre-trained Models for Natural Language Processing: A Survey. *Science China Technological Sciences*, *63*(10), 1872–1897.

Radev, D. R., Hovy, E. & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational linguistics*, *28*(4), 399–408.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Reimers, N. & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rumelhart, D., Hinton, G. & Williams, R. (1986). Learning representations by back-propagating errors.

Russell, S. & Norvig, P. (2009). *Artificial Intelligence: A Modern Approach* (3rd). Prentice Hall Press.

Sarkar, D., Bali, R. & Sharma, T. (2017). *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems* (1st). Apress.

Schroff, F., Kalenichenko, D. & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Schütze, H., Manning, C. D. & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39). Cambridge University Press Cambridge.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. & Batra, D. (2017). *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization* (tech. rep.).

Shahapure, K. R. & Nicholas, C. (2020). Cluster quality analysis using silhouette score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 747–748.

Sheikholeslami, S. (2019). Ablation programming for machine learning.

Stang, H. J. & Sollid, I. (2020). Biomedical Text Summarization Using Pre-trained Language Models.

Su, D., Xu, Y., Winata, G. I., Xu, P., Kim, H., Liu, Z. & Fung, P. (2019). Generalizing question answering system with pre-trained language model fine-tuning. *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, 203–211.

Su, J., Cao, J., Liu, W. & Ou, Y. (2021). Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Subramanian, S., Li, R., Pilault, J. & Pal, C. (2019). On extractive and abstractive neural document summarization with transformer language models. *arXiv preprint arXiv:1909.03186*.

Sutskever, I., Vinyals, O. & Le, Q. V. (2014). *Sequence to Sequence Learning with Neural Networks* (tech. rep.).

Tan, A.-H. et al. (1999). Text mining: The state of the art and the challenges. *Proceedings of the pakdd 1999 workshop on knowledge disocovery from advanced databases*, *8*, 65–70.

Tan, P.-N., Steinbach, M. & Kumar, V. (2006). *Introduction to data mining*. Pearson Education India.

Th, M., Sahu, S. & Anand, A. (2015). Evaluating distributed word representations for capturing semantics of biomedical concepts. *Proceedings of BioNLP@IJCNLP*, 158–163.

Tretyak, V. (2020). Combination of abstractive and extractive approaches for summarization of long scientific texts. *arXiv preprint arXiv:2006.05354*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan & R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 5998–6008). Curran Associates, Inc.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*(4), 395–416.

Waheeb, S. A., Khan, N. A., Chen, B. & Shang, X. (2020). Multidocument arabic text summarization based on clustering and word2vec to reduce redundancy. *Information*, *11*(2), 59.

Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A. & Setiadi, D. R. I. M. (2020). Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*.

Workman, T. E., Fiszman, M. & Hurdle, J. F. (2012). Text summarization as a decision support aid. *BMC Medical Informatics and Decision Making*, *12*(1), 1–12.

Yan, J. (2009). Text representation. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of database systems* (pp. 3069–3072). Springer US.

Yang, C., Rangarajan, A. & Ranka, S. (2018). Visual Explanations From Deep 3D Convolutional Neural Networks for Alzheimer's Disease Classification. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, *2018*, 1571–1580.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv*.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. & Hovy, E. (2016). Hierarchical attention networks for document classification. *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1480–1489.

Yildirim, F. B. & Sarikcioglu, L. (2007). Marie jean pierre flourens (1794–1867): An extraordinary scientist of his time. *Journal of Neurology, Neurosurgery & Psychiatry*, *78*(8), 852–852.

Zaheer, M., Guruganesh, G., Dubey, A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L. et al. (2020). Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.

Zhang, J., Zhao, Y., Saleh, M. & Liu, P. J. (2019). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv*, 119.

Zhao, J., Liu, M., Gao, L., Jin, Y., Du, L., Zhao, H., Zhang, H. & Haffari, G. (2020). Summpip: Unsupervised multi-document summarization with sentence graph compression. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1949–1952.

Zhao, Y., Saleh, M. & Liu, P. J. (2020). Seal: Segment-wise extractive-abstractive long-form text summarization. *arXiv preprint arXiv:2006.10213*.

Zheng, C., Zhang, K., Wang, H. J. & Fan, L. (2020). Topic-Aware Abstractive Text Summarization.

Appendix A

Gold summaries

Example A.1: Gold summary - "High ROUGE-1F score and moderate-redundant"

Infants' spontaneous and voluntary movements mirror developmental integrity of brain networks since they require coordinated activation of multiple sites in the central nervous system. Accordingly, early detection of infants with atypical motor development holds promise for recognizing those infants who are at risk for a wide range of neurodevelopmental disorders (e.g., cerebral palsy, autism spectrum disorders). Previously, novel wearable technology has shown promise for offering efficient, scalable and automated methods for movement assessment in adults. Here, we describe the development of an infant wearable, a multi-sensor smart jumpsuit that allows mobile accelerometer and gyroscope data collection during movements. Using this suit, we first recorded play sessions of 22 typically developing infants of approximately 7 months of age. These data were manually annotated for infant posture and movement based on video recordings of the sessions, and using a novel annotation scheme specifically designed to assess the overall movement pattern of infants in the given age group. A machine learning algorithm, based on deep convolutional neural networks (CNNs) was then trained for automatic detection of posture and movement classes using the data and annotations. Our experiments show that the setup can be used for quantitative tracking of infant movement activities with a human equivalent accuracy, i.e., it meets the human inter-rater agreement levels in infant posture and movement classification. We also quantify the ambiguity of human observers in analyzing infant movements, and propose a method for utilizing this uncertainty for performance improvements in training of the automated classifier. Comparison of different sensor configurations also shows that four-limb recording leads to the best performance in posture and movement classification.

Introduction: Clinical guidelines recommend using neuroimaging, Prechtls' General Movements Assessment (GMA), and Hammersmith Infant Neurological Examination (HINE) to diagnose cerebral palsy (CP) in infancy. Previous studies provided excellent sensitivity and specificity for each test in isolation, but no study has examined the pooled predictive power for early diagnosis. Methods: We performed a retrospective case-control study of 441 high-risk infants born between 2003 and 2014, from three Italian hospitals. Infants with either a normal outcome, mild disability, or CP at two years, were matched for birth year, gender, and gestational age. Three-month HINE, GMA, and neuroimaging were retrieved from medical records. Logistic regression was conducted with log-likelihood and used to determine the model fit and Area Under the Curve (AUC) for accuracy. Results: Sensitivity and specificity for detecting CP were 88% and 62% for three-month HINE, 95% and 97% for absent fidgety GMs, and 79% and 99% for neuroimaging. The combined predictive power of all three assessments gave sensitivity and specificity values of 97.86% and 99.22% (PPV 98.56%, NPV 98.84%). Conclusion: CP can be accurately detected in high-risk infants when these test findings triangulate. Clinical implementation of these tools is likely to reduce the average age when CP is diagnosed, and intervention is started.

Example A.2: Gold Summary - "Highest ROUGE-1F-scored (23.19) non-redundant summary"

General movements (GMs) are spontaneous movements of infants up to five months post-term involving the whole body varying in sequence, speed, and amplitude. The assessment of GMs has shown its importance for identifying infants at risk for neuromotor deficits, especially for the detection of cerebral palsy. As the assessment is based on videos of the infant that are rated by trained professionals, the method is time-consuming and expensive. Therefore, approaches based on Artificial Intelligence have gained significantly increased attention in the last years. In this article, we systematically analyze and discuss the main design features of all existing technological approaches seeking to transfer the Prechtl's assessment of general movements from an individual visual perception to computer-based analysis. After identifying their shared shortcomings, we explain the methodological reasons for their limited practical performance and classification rates. As a conclusion of our literature study, we conceptually propose a methodological solution to the defined problem based on the groundbreaking innovation in the area of Deep Learning.
Introduction New international clinical practice guidelines exist for identifying infants at high risk of cerebral palsy (CP) earlier: between 12 to 24 weeks corrected age, significantly earlier than previous diagnosis windows in Aus-

tralia at 19 months. The earlier detection of infants at high risk of CP creates an opportunity for earlier intervention. The quality of the parent-infant relationship impacts various child outcomes, and is leveraged in other forms of intervention. This paper presents the protocol of a randomised controlled trial of an online parent support programme, Early Parenting Acceptance and Commitment Therapy (Early PACT) for families of infants identified as at high risk of CP. We predict that participating in the Early PACT programme will be associated with improvements in the parent-infant relationship, in parent mental health and well-being as well as infant behaviour and quality of life.
Methods and analysis This study aims to recruit 60 parents of infants (0 to 2 years old corrected age) diagnosed with CP or identified as at high risk of having CP. Participants will be randomly allocated to one of two groups: Early PACT or waitlist control (1:1). Early PACT is an online parent support programme grounded in Acceptance and Commitment Therapy (ACT). It is delivered as a course on an open source course management system called edX. Early PACT is designed to support parental adjustment and parent-infant relationship around the time of early diagnosis. Assessments will be conducted at baseline, following completion of Early PACT and at 6-month follow-up (retention). The primary outcome will be the quality of parent-child interactions as measured by the Emotional Availability Scale. Standard analysis methods for randomised controlled trial will be used to make comparisons between the two groups (Early PACT and waitlist control). Retention of effects will be examined at 6-month follow-up.
Ethics and dissemination This study is approved through appropriate Australian and New Zealand ethics committees (see in text) with parents providing written informed consent. Findings from this trial will be disseminated through peer-reviewed journal publications and conference presentations.
Trial registration details This trial has been prospectively registered on 12 June 2018 to present (ongoing) with the Australian New Zealand Clinical Trials Registry (ACTRN12618000986279); https://www.anzctr.org.au/Trial/Registration/TrialReview.aspx?id=374896

Example A.3: Gold summary - "Low ROUGE-1F (14.83) and highly redundant"

To improve the neurodevelopmental outcome in infants with high grade intraventricular haemorrhage and cramped-synchronised (CS) general movements (GMs). Four very preterm infants with intraventricular haemorrhage grade III (n=3) or intraventricular haemorrhage with apparent periventricular haemorrhagic infarction (n=1) were diagnosed with CS GMs at 33 to 35 weeks postmenstrual age. A few days later MIT-PB [Movement Imitation

Therapy for Preterm Babies], an early intervention programme, was commenced: the instant an infant showed CS movements, the therapist intervened by gently guiding the infant's limbs so as to manoeuvre and smoothen the movements, thereby imitating normal GM sequences as closely as possible (at least for 10 min, 5 times a day, with increasing frequency over a period of 10 to 12 weeks). After a period of consistent CS GMs, the movements improved. At 14 weeks postterm age, the age specific GM pattern, fidgety movements, were normal in three infants, one infant had abnormal fidgety movements. At preschool age, all participants had a normal neurodevelopmental outcome. This report on four cases demonstrates that mimicking normal and variable GM sequences might have a positive cascading effect on neurodevelopment. The results need to be interpreted with caution and replication studies on larger samples are warranted. Nonetheless, this innovative approach may represent a first step into a new intervention strategy.

Background Prediction of long-term neurodevelopmental outcomes remains an elusive goal for neonatology. Clinical and socioeconomic markers have not proven to be adequately reliable. The limitation in prognostication includes those term and late-preterm infants born with neonatal encephalopathy. The General Movements Assessment tool by Prechtl has demonstrated reliability for identifying infants at risk for neuromotor impairment. This tool is non-invasive and cost-effective. The purpose of this study is to identify the published literature on how this tool applies to the prediction of cerebral palsy in term and late-preterm infants diagnosed with neonatal encephalopathy and so detect the research gaps.

Methods We will conduct a systematic scoping review for data on sensitivity, specificity, positive, and negative predictive value and describe the strengths and limitations of the results. This review will consider studies that included infants more than or equal to 34 + 0 weeks gestational age, diagnosed with neonatal encephalopathy, with a General Movements Assessment done between birth to six months of life and an assessment for cerebral palsy by at least 2 years of age. Experimental and quasi-experimental study designs including randomized controlled trials, non-randomized controlled trials, before and after studies, interrupted time-series studies and systematic reviews will be considered. Case reports, case series, case control, and cross-sectional studies will be included. Text, opinion papers, and animal studies will not be considered for inclusion in this scoping review as this is a highly specific and medical topic. Studies in the English language only will be considered. Studies published from at least 1970 will be included as this is around the time when the General Movements Assessment was first introduced in neonatology as a potential predictor of neuromotor outcomes. We will search five databases (MEDLINE, Embase, PsychINFO, Scopus, and CINAHL). Two reviewers will conduct all screening and data extraction in-

dependently. The articles will be categorized according to key findings and a critical appraisal performed.
Discussion The results of this review will guide future research to improve early identification and timely intervention in infants with neonatal encephalopathy at risk of neuromotor impairment.
Systematic review registration Title registration with Joanna Briggs Institute https://joannabriggs.org/ebp/systematic_review_register.

Helene J. Stang, Ingeborg S. Sollid

Master's thesis

NTNU
Norwegian University of
Science and Technology