Marie Kjellstrøm Thorkildsen

Supporting Explainability in Machine Learning Systems Using Text Summarization

Masteroppgave i Datateknologi
Veileder: Heri Ramampiaro
Juni 2020

Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

NTNU
Kunnskap for en bedre verden

Marie Kjellstrøm Thorkildsen

Supporting Explainability in Machine Learning Systems Using Text Summarization

Masteroppgave i Datateknologi
Veileder: Heri Ramampiaro
Juni 2020

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

NTNU
Norwegian University of
Science and Technology

Abstract

The applications of artificial intelligence have increased significantly in the last years, and its utility has been adopted in the medical domain. Machine learning has shown great potential for early diagnostication of cerebral palsy, where artificial intelligence is used to track and analyze the movements of an infant. For clinicians to use the predictions of high or low risk for cerebral palsy, the system must be able to clarify the reason for the given predictions. This explanation must then be verified by clinicians before determining a diagnosis. However, this is not an easy task, since the currently used medical search engines are cumbersome and returns too many search results.

This master's thesis proposes a search system for medical articles consisting of several composed features. This system assists clinicians in a fast review of relevant domain knowledge by retrieving, clustering and summarizing the most important information. For constructing these summaries, we propose a novel method that extracts relevant sentences by identifying important concepts in the documents and using word frequencies to adjust the importance of each sentence. This method proved to be successful, and it performed significantly better than the baseline methods. Further, our experiments showed that the improvement was even larger for scientific papers in general, and our research is therefore a valuable contribution to the summarization domain.

Sammendrag

Bruksområdene for kunstig intelligens har vokst betraktelig de siste årene, og dette har også hatt stor nytteverdi for det medisinske domenet. Maskinlæring har vist seg å være nyttig for tidlig diagnostisering av cerebral parese, hvor kunstig intelligens brukes til å følge og analysere bevegelsene til et spedbarn. For at klinikere skal få bruk for prediksjonene av høy eller lav risiko for cerebral parese, må systemet være i stand til å belyse bakgrunnen for prediksjonene som gis. Den gitte forklaringen må igjen verifiseres av klinikere før en eventuell diagnose kan stilles. Dette er imidlertid ikke en enkel oppgave, da dagens søkesystemer for medisinske artikler er tungvinte og returnerer en for stor mengde resultater.

Denne masteroppgaven foreslår et nytt søkesystem for medisinske artikler basert på flere sammensatte funksjoner. Dette systemet bistår klinikere i en rask gjennomgang av relevant fagkunnskap ved å hente ut, gruppere og gi et kort sammendrag av den viktigste informasjonen. For å konstruere disse sammendragene foreslår vi en ny metode, som trekker ut relevante setninger ved å identifisere viktige konsepter i dokumentene og å bruke ordfrekvenser til å justere viktigheten av hver enkelt setning. Denne metoden viste seg å være vellykket, og den presterte betydelig bedre enn de grunnleggende metodene. Videre viste eksperimentene at forbedringen var enda større for generelle vitenskapelige artikler, og forskningen er dermed et verdifullt bidrag til sammendragsdomenet.

Preface

This thesis is written for the Department of Computer Science at Norwegian University of Science and Technology. The research was conducted by Marie Kjellstrøm Thorkildsen during the spring of 2020 as the final assignment for a five year Master of Science degree. An associated specialization project was conducted during the autumn of 2019, and as that work is still relevant, some parts of it have been included in this thesis.

The thesis contributes to a large cross-department research project by Norwegian University of Science and Technology and St. Olavs University Hospital. The supervisor of this thesis has been Professor Heri Ramampiaro at the Department of Computer Science.

Acknowledgements

First, I am profoundly grateful to Heri Ramampiaro for giving me the opportunity to work on this exciting project. He has been my supervisor for the one year duration of this project, and I can not thank him enough for all his support and advice. He has given me the freedom to choose my own path, but also helped me when I needed guidance along the way.

Also, a special thanks to Researcher Lars Adde for sharing his deep knowledge of the domain and insights from a clinical perspective. These contributions have been of great help to understand the problem and find potential solutions.

Finally, I would like to express my gratitude to my boyfriend for his valuable feedback and support during this thesis, and to my family and friends for cheering for me and keeping me motivated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for the Project

Cerebral palsy is the most common physical disability in children, and in Norway 120 to 150 children get affected by this disorder every year [1]. The physical and cognitive challenges caused by cerebral palsy have negative impacts on the patient's life, but early diagnosis and custom treatment can significantly improve the life quality. This is achievable by using a method that evaluates the movements of an infant, which enables diagnosis at an age between 9 and 18 weeks. As this method has been highly successful, the assessment procedure has been digitalized using Artificial Intelligence (AI) to track and evaluate the infant's movements. This AI tool is called In-Motion, and it aims at being a useful supplement to the clinicians' own expertise.

However, when using a machine learning system for diagnosis purposes, such as In-Motion, the medical experts using this system must be able to understand how the algorithm "reasons". To achieve this, the system must be transparent, meaning that it can explain the reasons for giving a certain prediction. In fact, GDPR requires machine learning systems to deliver such explanations if the prediction significantly impacts the patient's life [2]. After the system has given an explanation, the reasoning must be validated by a human, usually by finding and reading related research. This could be performed by manually searching for articles using a medical search engine, but the system that is used by most medical experts today is not suitable for validating a machine-generated prediction. The problems with this search engine are that it returns too many results, and that there is no way to quickly digest the content of each result other than reading the article.

In summary, AI-based methods are not very useful for clinical purposes if they can not provide enough transparency for medical experts to verify the predictions. This means that even though the In-Motion system shows great promise and potential, the problems related to explainability need to be solved before it can be used extensively.

1.2 Problem Statement

As described in the previous section, not receiving an explanation of how a prediction was made is a problem for the medical personnel who use the In-Motion system. The goal of this thesis is therefore to investigate how text mining can be used to assist medical personnel with understanding and verifying predictions generated by the In-Motion system. One way to accomplish this goal is to build a text mining tool that can assist in answering questions regarding predictions from In-Motion.

For such a text mining to fully integrate with In-Motion, it requires the AI system to return the movement characteristics that were decisive for the given prediction. These characteristics can then be used as input to a text mining tool, either by the In-Motion system itself or by medical personnel constructing a query that reflects what they are looking for. By considering the set of characteristics as query keywords, a text mining tool should retrieve relevant knowledge within the cerebral palsy domain in an attempt to help the user to either verify or discard the AI-generated prediction. An advantage of basing the text mining system on keyword inputs is that it can be combined with an AI method that is considered a "black-box" approach. More specifically, the system does not need to know the internals of the AI and how it "reasons", but can instead create an explanation based on the AI's output.

A text mining tool should retrieve knowledge for the user in such a way that it is easy and fast to digest, so that they can more quickly reach a conclusion regarding a diagnosis. In other words, the system should not be a traditional search engine, where the results are hyperlinks to documents, but instead it should be able to extract digestible knowledge for the user.

During our specialization project [3], which is considered to be part of this thesis, we developed a hypothesis for how such a tool could be built by combining existing text mining methods. First, relevant documents for the given input are retrieved from a corpus of documents. Further, these documents should be divided into an appropriate amount of clusters, and finally, each of these clusters should be summarized in order to extract the most essential knowledge from each set of related documents. We hypothesize that this proposed system, especially the summarization feature,

could provide the medical personnel the knowledge they need in a fast and accurate way to verify or discard the given predictions.

One of the main challenges is that such a system must process medical texts, while most text mining techniques are created to work well on general texts. To cope with this issue, some techniques may need to be modified to work better for texts from the biomedical domain. For instance, biomedical texts contain terminology that is specific to the domain, and general text mining approaches might not handle this terminology correctly.

1.2.1 Research Questions

To solve the problems stated above and to accomplish our goal of assisting medical personnel with understanding predictions, the following research questions were formulated:

RQ1: How can text summarization be used to support explainability in machine learning systems?

RQ2: Which adaptations must be applied for text mining techniques to work with the biomedical domain?

1.2.2 Scope

As this project is part of the larger research project In-Motion, further explained in Section 1.4, this thesis only focuses on the applicability of the described search engine. We are not concerned with how the prediction-decisive characteristics are output from the In-Motion tool, but develop a search system that is fully functional by itself and that can be connected to the In-Motion tool when the system outputs the proper data.

The system's speed is quite important for actual usage, as users do not want to wait long for results. However, since this thesis is limited in its time frame, making the system faster by tuning the text mining techniques is not considered a priority. Nevertheless, we still strive to deliver a system that is fast for the end-user.

1.3 Research Approach

This thesis was initiated with a quantitative study, where relevant literature was collected to get an overview of the research within the text mining field. To fully understand the context of the project challenges, we explored the fields of Explainable Artificial Intelligence and existing information retrieval systems, and investigated what makes existing systems insufficient

for our purpose. However, the main focus was to explore text analysis and to research potential mining methods and their performance.

In our specialization project [3], we experimented with the fundamentals of our search system. More specifically, we investigated which information retrieval models are suited for the medical domain and which adaptations work well for processing medical texts. Further, the work of this thesis has focused on using the retrieved documents to process the content and present it to medical personnel in an effective and understandable manner. To achieve this, we investigated how important content in articles could be extracted and presented in short summaries to the user.

1.4 Context

This thesis is part of an extensive research project by the Norwegian University of Science and Technology (NTNU) and St. Olavs University Hospital. The project team consists of researchers from both the Department of Clinical and Molecular Medicine, the Department of Neuromedicine and Movement Science and the AI-LAB at NTNU. Unlike most of the work on the In-Motion project, this thesis does not focus on movement tracking of infants or prediction of risk for cerebral palsy, but instead tackles the problem of explainability for these predictions.

1.5 Contributions

For In-Motion to be recognized as a reliable tool, it must provide transparency and be able to clarify the reasoning behind the given predictions. By achieving this, the medical personnel can get insight into how a decision was made, and be able to verify this reasoning before determining a diagnosis. The main contribution of this thesis is to investigate how text analysis can be used as a decision-supportive tool. We build a proof of concept tool that can be used by medical personnel to retrieve relevant knowledge and research within the cerebral palsy domain. This enables clinicians to be more confident in acknowledging the prediction as valid or understand why the prediction is made on the wrong basis.

1.6 Outline of Dissertation

As mentioned earlier, the specialization project [3] is a part of this master's thesis, and much of the work conducted during that project is still highly relevant. For convenience to the reader, this research is included or adapted in parts of chapters 2, 3 and 4.

The thesis contains the following chapters:

- **Chapter 1 – Introduction** describes the motivation behind this thesis, our research goals and how we choose to approach the problem.
- **Chapter 2 – Theoretical Background** presents the relevant background knowledge for this thesis. Additionally, this chapter describes the text mining techniques we plan to base our proposed solution upon.
- **Chapter 3 – State of the Art** discusses research related to our final search system, and evaluates existing information retrieval systems within the medical domain.
- **Chapter 4 – Approach** describes our proposed system used to support explainability in the In-Motion AI.
- **Chapter 5 – Results** presents the results of our experiments, and contains comparisons to other text mining approaches.
- **Chapter 6 – Discussion** evaluates the findings from the previous chapter, and discusses the usefulness of our applied techniques.
- **Chapter 7 – Conclusion and Future Work** concludes our research, and describes our ideas for future work based on this thesis.

Chapter 2

Theoretical Background

This chapter presents the relevant background knowledge for our thesis. First, cerebral palsy and one of its diagnosis procedures are described. Further, we present Explainable Artificial Intelligence, why it is important and some approaches for solving the challenges in this domain. Finally, we describe three text mining techniques that our implementation is based on. Each technique contains an introduction of its purpose, followed by a description of potential approaches to implement it. Lastly, we show how the performance of each technique can be evaluated.

2.1 Cerebral Palsy

Cerebral palsy (CP) is a neurological condition caused by damage to the brain, and it results in motoral and cognitive difficulties. The brain damage can occur anytime from the fetal period until an age of approximately two years [1]. Cerebral palsy affects the signals sent in both directions between the brain and the muscles. Disturbances in signals sent from the brain to the muscles cause uncontrolled movements, while disturbances in the other direction cause unawareness of where the muscles are located spatially. Cerebral palsy has a wide spectrum of potential symptoms, where delayed milestones is one of the most prominent for infants [4]. Asymmetrical movements and spasticity are also important markers, where the latter is caused by increased muscle tone. The extent of movement difficulties is divided into five categories [5], which varies from person to person. Some might only experience mild degrees of spasticity, i.e. muscle stiffness, while others might have extensive challenges and are dependent on wheelchairs.

Cerebral palsy is an incurable disorder, and proper treatment and care are therefore essential for affected persons' life quality [1]. The follow-up and potential diagnostication of a high-risk patient is performed by the

hospitals, and they are checking several areas of development, for instance movements, language and perception. A cerebral palsy diagnosis is usually given when the child is between one and two years old. However, the life quality of the patient can be greatly improved if receiving a diagnose at an earlier stadium, preferably between an age of two to five months [6]. This is because their brains are still malleable at this point, which means that the treatment can have a larger impact.

2.2 General Movement Assessment

General Movement Assessment (GMA) is a method for identifying cerebral palsy at an early age, preferably around two or three months. GMA is performed by studying the movements of an infant, and is used to determine whether there is a high risk of cerebral palsy or not. GMA was developed by Prechtl [7], and it has been highly successful. GMA plays an important role in helping with early identification of cerebral palsy, especially since it was adopted as an international guideline in 2017 [5, 8].

The examination of the infant is performed by analyzing its movement while lying on the back. These movements are preferably recorded by video at the infants' home, where the behavior of the infant is as natural as possible [8]. The main objective of GMA is to identify whether the infant is able to express *fidgety movements* or not [9]. Fidgety movements are small movements of various speeds and directions, which are usually present at an age of 3 to 5 months. Lack of fidgety movements is a good indicator of high risk for cerebral palsy [5].

GMA is based on Gestalt perception [10], which in this scenario means that trained medical personnel perform a visual observation and assessment of the infants' movements. One challenge with GMA, and Gestalt perception in particular, is that the examiners may bring their subjectivity into the diagnostication, which has caused skepticism to whether or not this can be recognized as a standard tool for identifying CP [9].

GMA has been extensively used by St. Olavs Hospital in Trondheim to estimate the likeliness of developing cerebral palsy [9], and is combined with other examinations like ultrasound, MRI and neurologic evaluation. However, in the last few years, St. Olavs Hospital has initiated a research project in collaboration with the Norwegian University of Science and Technology (NTNU) to create a software tool based on the GMA techniques [6]. This tool is called *In-Motion*, and is developed by medical experts and computer scientists from the two research institutions.

2.2.1 In-Motion

In-Motion [11] is an application that can be used on smartphones. The app is used to create a recording of the infant's movements while it is lying on its back, which is then analyzed by a human and a machine to determine whether or not the infant has a high risk of developing cerebral palsy. The human expert determines if there is a high or low risk, and uses the machine-generated prediction to support their decision.

The In-Motion tool uses Artificial Intelligence (AI) to analyze the recording and predict the likeliness of the infant developing cerebral palsy [6]. The AI model used by In-Motion is trained on thousands of videos that were gathered in a research project spanning over a decade. In order to analyze the movements of the infant, several parts of the body are identified and automatically tracked by using the method developed by Groos and Aurlien [12]. An example of this automatic tracking is shown in Figure 2.1.

Figure 2.1: This figure shows usage of the method developed by Groos and Aurlien [12], and illustrates how the body parts of an infant are marked by the In-Motion tool. Once these body parts are identified, their movements can be automatically tracked and analyzed.

One of the challenges with using AI for this purpose, as described in Section 2.3, is that the system needs to provide transparency because it is used for diagnostication. As mentioned in the introduction, the goal of our thesis

is to support explainability in the In-Motion AI.

Since the In-Motion tool is an app for a smartphone, it can be used anywhere and by anyone. This feature leads to several advantages, an important one being that the In-Motion tool can make the GMA diagnosis method available to a larger population, who may not have a hospital with the right expertise close by [8]. This could make a big difference for these families, as their children can receive a diagnosis and thereby proper treatment earlier. Even for families who live near a hospital with the right expertise, the tool can allow them to take fewer trips to the hospital. Additionally, since recordings can be made at home, the infant is more likely to be calm during the recording, making its movements more natural.

2.3 Explainable Artificial Intelligence

The study of relevant background knowledge for this thesis was done during the specialization project [3], which, as mentioned earlier, can be considered being part of the work for this thesis. Specifically, the information regarding Explainable Artificial Intelligence from our specialization project is still highly relevant for this thesis, and has for convenience to the reader been included in this section.

Artificial Intelligence (AI) has one essential limitation: the inability to explain the results it generates [13]. A large number of current AI systems provide no transparency or explanation of how decisions are made. This has raised interest in a relatively new field of study called Explainable Artificial Intelligence (XAI). The U.S. Defense Advanced Research Project Agency (DARPA) made XAI one of its focus areas in 2017 [14].

The goal of XAI is to provide explanations for the reasoning behind given predictions or classifications. An explanation usually describes which features of the input lead to the given output in such terms that a human can understand. XAI is mainly used for ensuring the correctness of an AI system, i.e. that the system's prediction is based on the expected features, but it can also be used to find previously unknown patterns and improve the AI system [15].

There exists very little research about explainable AI and text[1]. This is also a difficult task to solve, as it has two significant challenges [16]. First, a textual explanation will only be able to indirectly explain the internal logic of the AI. Second, these explanations are not very useful for discovering false predictions.

[1]Multiple searches on Google and Google Scholar were conducted to retrieve research describing how explanations can be given in a textual format.

The EU General Data Protection Regulation (GDPR) introduced a major challenge for existing machine learning systems: when receiving a prediction that significantly affects the user, he can demand an explanation of how the system's decision was made [2]. It should be noted that machine learning systems are not required to provide explanations unsolicited, but must be able to explain how the results were achieved on demand.

Explainable Artificial Intelligence is needed in many industries, but the transparency it provides is particularly vital in healthcare [17]. Doctors, physiotherapists and other specialists need to understand the system's reasoning in order to trust it before using it to decide a proper diagnosis.

2.3.1 Explainable AI Models

Explainable AI systems can either be *post-hoc* or *ante-hoc* [18]. Post-hoc systems consider the AI model as a black box, which means the internals of the model do not need to be known. Post-hoc explanations can be generated on a case-by-case basis, instead of having an explanation for the whole system's behavior. With ante-hoc systems, on the other hand, the explanation is integrated in the AI system itself. Post-hoc systems are useful because they can be integrated with existing classifiers, rather than having to re-build the classifier with explainability in mind. A selection of post-hoc systems are described in the sections below.

Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanations (LIME) [19] is a technique for explaining predictions given by machine learning systems. LIME aims to be compatible with all types of classifiers, and this is achieved by creating a human interpretable model that reflects the behavior of the given classifier.

For text classification, an explanation can be a binary vector specifying whether words are present in the text or not [19]. The number of words contributing to the prediction could be large, and LIME ensures an interpretable explanation by limiting the number of words displayed to the user. The limit should not exceed the maximum amount of words that the user can handle.

LIME has several significant advantages [13]. First, one can generate explanations with the same interpretable model, independently of which machine learning model is used. Second, LIME works on both images, text, and tabular data. Most other systems have issues with at least one of these. Additionally, an experiment [19] was conducted to investigate whether users could differentiate between correct predictions and predictions made on a

wrong basis. After receiving explanations from LIME, almost all users were able to identify the basis behind the classification, compared to less than 50% before receiving explanations. The experiment shows that LIME provides insight into the classifier's behavior, and enables users to distinguish between predictions made on a correct basis and those who are not.

Layerwise Relevance Propagation

Layerwise Relevance Propagation (LRP) [15] is another technique for explaining the reasoning behind a prediction, and moves backwards in a neural network to identify the relevant contributions to the result. In the case of classifying pictures, LRP would identify relevant pixels that contributed to the given prediction.

An experiment [15] was conducted with LRP in the context of text document classification. The model was given text documents to analyze and classify, and was then to give an explanation for the reasoning behind the classification. LRP tries to distinguish between words that contribute to the classifications and words that have a negative effect. The output of the experiment was the text annotated with a so-called *heatmap*, which marked positive contributions as red and negative contributions as blue. The experiment showed that heatmaps are useful for explaining a text classification, and LRP was proven to perform much better than a similar method that could not differentiate between positive and negative contributions.

Black Box Explanations through Transparent Approximations

Black Box Explanations through Transparent Approximations (BETA) [20] tries to learn some *decision sets* which mimic and explain the behavior of the AI they analyze. BETA also allows the user to decide which input should be analyzed, and thus enables the user to explore different aspects of the AI.

A user study [20] was conducted to explore how well users understood the behavior of an AI model in different scenarios. The experiment shows that users using explanations from BETA were significantly more accurate than users using Interpretable Decision Sets (IDS)[2] and Bayesian Decision Lists (BDL)[3], which are other XAI models. Furthermore, BETA users were also

[2]LAKKARAJU, Himabindu; BACH, Stephen H.; LESKOVEC, Jure. Interpretable decision sets: A joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016. p. 1675-1684.

[3]LETHAM, Benjamin, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics, 2015, 9.3: 1350-1371.

50% and 130% faster to answer the given questions than IDS and BDL users, respectively.

2.4 Information Retrieval

Information retrieval is the task of analyzing documents and identifying which are relevant for a given query [21]. The documents can be binarily classified as relevant and non-relevant, but most often they are given a score and ranked based on how relevant they are to the given query. Queries can be written in several ways, for instance in a boolean fashion or by using important terms, often referred to as "keywords". A good retrieval system should be able to retrieve as many relevant documents as possible, but it should at the same time avoid retrieving non-relevant documents.

The retrieval ability of a system can in some cases be improved by allowing feedback from the user of whether the content is relevant or not [21]. When working with user feedback, the content of the documents marked as relevant and non-relevant is analyzed. This knowledge is then used to add relevant words to the query the next time it is executed. User feedback is especially useful if the queries are written in a complex manner, for instance if the query is formulated as a question.

Preprocessing of queries and documents is an essential procedure for achieving high retrieval performance [21]. The most common preprocessing technique is to remove *stopwords* from the texts. Stopwords are commonly used words that do not add much value to the text, for instance "for", "to" and "the". Another technique is *stemming*, which removes prefixes and suffixes from words to get the most basic form of the word. This technique is created to handle the fact that a word can be written in several variations, for instance different verb tenses. Stemming has shown to improve the recall value significantly, but it might affect precision negatively. This should be taken into consideration when developing a retrieval system. Other common techniques worth mentioning are lemmatization, lower- or uppercasing of words and removal of numbers and special characters [21, 22].

2.4.1 Ranking Models

Document retrieval and ranking is an extensively researched area, and a wide spectrum of models have been developed over the years. To exemplify this diversity, we describe the approaches of Boolean Model, Language Model, Vector Space Model and Okapi BM25 below, and our descriptions are based on the work by Liu [21]. These models have taken very different approaches to solving the retrieval challenge, and each has their strengths and weaknesses. For this reason, no model is best in every case, as each

case has unique challenges. It should be noted that the models described below are the baseline approaches, and small changes can greatly affect their performances.

Boolean Model

The Boolean Model is a very simple approach, and in contrast to the models described below, Boolean Model classifies the documents as match or non-match rather than ranking them after their relevance to the query. The Boolean Model uses boolean algebra to determine whether a document is a match or not, and this also requires the query to be written in a boolean manner with *AND*, *OR* and *NOT*. Since the query terms are combined in a boolean manner, the documents must be an exact match, i.e. fulfill all query criteria, for it to be classified as a match.

Language Model

The Language Model is a statistical and probabilistic retrieval approach. Instead of using the query to determine whether documents are relevant or not, like the Vector Space Model and Okapi BM25 described below, Language Model calculates the probability of generating the query given the documents. This is performed by constructing a language model for each document, and then calculating the probability that each of them generates the given query. The documents are then ranked based on their probability score.

Language Model ranking can be performed using different forms of n-grams, i.e. n subsequent words. However, the most common n to use in Language Model is $n = 1$, also known as unigrams. To calculate the probability of a document to generate a query, Language Model computes the probability of each query term occurring in the document, and then multiplies all of these probabilities. However, this can cause problems since some word probabilities might be zero. To deal with this problem, Language Model uses *smoothing*, which normalizes the probabilities by increasing very low and zero values and decreasing very high values.

Vector Space Model

The Vector Space Model (VSM) is one of the most commonly used models within information retrieval. VSM is based on Term Frequency Inverse Document Frequency (TF-IDF), which analyzes the occurrences of each term and punish commonly used words. Correspondingly, it also rewards rarely used words, as they might be more important.

To rank the document, VSM measures the distance between the documents

and a query. This is performed by transforming the documents and the query to vector format, and then calculating the cosine similarity between the vectors. After measuring the cosine similarity between all document and query pairs, the documents are ranked based on their similarity score. The higher the similarity score, the higher the raking. This is a quite different approach than Boolean Model, which only classifies documents as match or non-match.

Okapi BM25

Okapi BM25 is a probabilistic ranking model, and it calculates the probability that a document d is relevant given a query q. For short queries, Okapi BM25 generally performs better than cosine similarity. BM25 is based on TF-IDF [22], which, as mentioned above, rewards rare terms and punishes commonly used terms. BM25 is a "bag-of-words" model, which means that it evaluates each word independently, and does not consider adjacent words. The following formula is used to calculate the BM25-score for a given document-query pair [21, 23]:

$$okapi(d_j, q) = \sum_{t_i \in q, d_j} IDF(t_i) \cdot \frac{(k_1 + 1) f_{ij}}{k_1(1 - b + b\frac{dl_j}{avdl}) + f_{ij}} \cdot \frac{(k_2 + 1) f_{iq}}{k_2 + f_{iq}} \quad (2.1)$$

dlj is the length of document j, while $avdl$ is the average length of all documents. f_{ij} and f_{iq} are term frequencies for term i in document j and query q, respectively. k_1, k_2 and b are parameters, and the values are usually $k_1 \in [1, 2]$, $k_2 \in [1, 1000]$ and $b = 0.75$.

2.4.2 Evaluation

Two commonly used measures for evaluating ranking models are *precision* and *recall* [21]. Precision calculates the ratio of retrieved documents that are relevant, and can be computed by using Equation 2.2. Recall calculates the ratio of relevant documents that are retrieved, and can be computed by using Equation 2.3.

$$Precision = \frac{Number\ of\ relevant\ documents\ in\ hitlist}{Number\ of\ documents\ in\ hitlist} \quad (2.2)$$

$$Recall = \frac{Number\ of\ relevant\ documents\ in\ hitlist}{Number\ of\ relevant\ documents\ in\ total} \quad (2.3)$$

The ideal system would achieve high performance on both precision and recall, but in reality, focus on one metric usually comes at the expense of

performing worse on the other metric [21]. This phenomenon is referred to as the "trade-off" between precision and recall, and can be evaluated using *F-measure*. This metric takes both precision and recall into account, which can be used to get a better indicator of the system's performance as a whole. F-measure is constructed such that a low precision or recall value will decrease the total score. In other words, one cannot achieve a satisfactory F-measure score without scoring well on both metrics. F-measure can be computed by using Equation 2.4.

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.4}$$

2.5 Unsupervised Learning

Unsupervised learning, also known as *clustering*, has as its objective to group items together such that the items within each *cluster* are similar or related [21]. Additionally, each cluster should be as distinct as possible, which means that the clusters should be clearly separated from each other. In other words, the items within a cluster should be as similar as possible, while at the same time be significantly different from items in other clusters. Figure 2.2 illustrates a clustering scenario where 16 data points are divided into three distinct clusters.

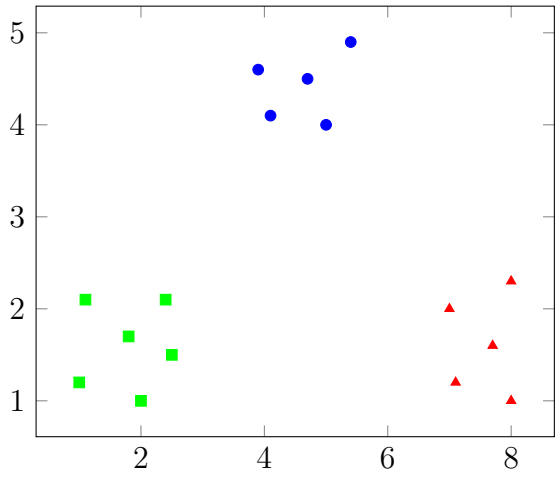

Figure 2.2: An example of 16 data points divided into three distinct clusters. The three clusters are marked with green squares, blue circles and red triangles.

Unsupervised learning algorithms do not need labeled data as input. Instead, all the data is analyzed to find patterns that are used to group items

into clusters. These patterns are unique for each dataset and are identified during clustering. It differs from supervised learning in that it does not require a trained classifier to cluster items, and therefore works on any dataset without modifications. The importance and popularity of unsupervised methods have increased significantly in the last years, as the quantity and size of datasets keep growing [21].

In the case of clustering documents, one measures how similar the content of a document is to the different clusters' content. The documents are then assigned to the cluster they have the highest textual similarity with. To compute this similarity, each document is usually converted to a vector that represents all the words in the document. Further, the similarity can be measured by calculating the distance or angle between the vectors.

2.5.1 Clustering Methods

Several clustering methods exist, and each of these has different focus areas and potential applications. The sections below will describe two significantly different approaches: the traditional partitioning method K-means and the more modern Topic Model approach.

K-means

K-means is one of the most used clustering algorithms [21], and is a partitional clustering method. This means that items are "partitioned" into k clusters, and that each item only can belong to one cluster. Because partitional clustering is computationally efficient, it performs very well for large-scale document sets [24].

The K-means procedure is as follows:

1. Set k random points as cluster centroids. Note that the value of k is determined a priori.
2. For each item: calculate the distance to each of the k centroids. Assign the item to the closest cluster.
3. For each cluster: move the centroid such that it is in the center of all points assigned to the cluster.
4. Repeat steps 2 and 3 until an exit condition is met. Usually, this condition is that the centroids stop changing position, that the assignment of clusters stays constant or that the clusters' *Sum of Squared Errors*, i.e. distances between the cluster centroids and its assigned items, changes less than a predetermined threshold.

One of the challenges with K-means is determining the optimal number of clusters. There are several options for determining this number, and two widely used methods are *the Elbow Method* and *the Silhouette Method* [25]. The Elbow method plots the Sum of Squared errors, described in Section 2.5.2, for a various number of clusters, and an example of this is visualized in Figure 2.3. The optimal number of clusters is where the graph has an "elbow", i.e. the point where increasing the number of clusters does not improve the Sum of Squared error as much as it did before the point. The Silhouette method measures the relation between items within the same cluster, compared to items in other clusters. This method is also used as an evaluation metric, and is described further in Section 2.5.2.

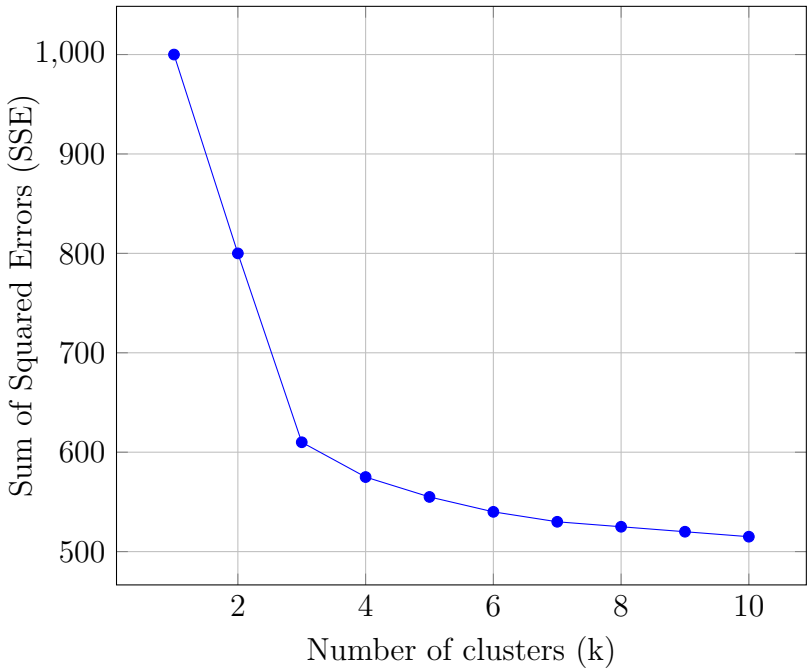

Figure 2.3: An example of how the elbow method plots Sum of Squared Errors (SSE) against the number of clusters. The elbow is where increasing the number of clusters does not result in a significant reduction of the SSE. In this case, the elbow is at $k = 3$.

Topic Modeling

Another alternative for clustering items is to use topic models [26], which is a newer and more modern clustering approach than K-means. It also differs from K-means in that it is a probabilistic method rather than partitional, which means that documents can be assigned to several clusters with different probabilities. Topic modeling is usually performed by using Probabilistic Latent Semantic Analysis (PLSA) or Latent Dirichlet Allocation

(LDA), where the latter is inspired by the former and usually performs better [26]. LDA is mostly used for summarization purposes, and is described in detail in Section 2.6.1.

An experiment [26] was conducted to evaluate the clustering capabilities of Topic Model. This experiment showed that it did not achieve as high performance as the more traditional method K-means. However, this experiment assumed that documents were assigned to the cluster with the highest probability, and this might have caused decreased performance since it is a probabilistic method and assumes that documents can be assigned to more clusters than one.

2.5.2 Evaluation

To evaluate how well clustering methods perform, one usually measures how distinct each cluster is and how related the items within each cluster are. Several methods exist for this purpose, but there are two main groups of evaluation: *labeled* metrics and *unlabeled* metrics [27]. These are also called *external validity measures* and *internal validation measures*, respectively.

Labeled metrics require the items to be classified before the clustering takes place. Note that these labels are not used in the clustering process itself, but as a ground truth to verify and evaluate the clustering abilities afterwards. *Purity*, *Rand index*, *Normalized mutual information* and *F-measure* are examples of labeled evaluation metrics [22]. All these metrics have in common that they compare the clustering to a "gold standard", i.e. the ideal classification of items.

Unlabeled metrics, on the other hand, do not require classified items to evaluate the clustering ability. Commonly used metrics for unlabeled data are *the Silhouette Method* [28] and *Sum of Squared Errors* (SSE) [29]. Each item's silhouette score calculates how similar it is to items within the same cluster versus how similar it is to items in other clusters. Silhouette outputs a number between -1 and 1, where a silhouette score of -1 implies that the items are placed in the wrong clusters, while a score of 1 means perfectly distinct clusters. If one achieves a silhouette score of 0, it means that clusters probably are overlapping and that items could have been placed in other nearby clusters. SSE uses Euclidean distance to measure the distance between an item and the centroid of its assigned cluster. The goal of SSE is to reach a score as close to zero as possible, since this means that the items within a cluster are very close to each other and therefore related. To evaluate a whole clustering run, this procedure is repeated for each item within each cluster. The final SSE score is the sum of each cluster's SSE score.

2.6 Text Summarization

Text summarization is the task of summarizing the most important content in a given text [30]. One significant strength of text summarization, if done well, is that it allows humans to digest the most important information of a document quickly by only reading a fraction of the content. This can be especially useful when performing a broad search, as it can help the user to understand the search results without spending too much time on each document. Summarization has several use cases, one of which is making it easier to review search results by summarizing the content of documents. Another use case for summarization is generating previews or summaries of news articles to allow users to digest and filter content faster.

In general, there are two variants of summarization: *extractive* and *abstractive* [31]. Extractive summarization is a selection of sentences from the document source, where the sentences are added to the summary unchanged. Abstractive summarization, on the other hand, generates sentences based on the knowledge in the document source. Thus, an abstractive-made summary is a summary of the important knowledge in the document, while an extractive-made summary is a subset of the most important sentences.

Summarization is a widely researched domain, but there has still not been found a generally good approach [31]. Summarization presents several challenges, for instance evaluation of summaries. Most evaluation metrics, as described in Section 2.6.2, require a gold standard summary for the constructed summary to be compared with. However, an ideal summary is hard to construct, and often requires manual work by humans. This is a time-consuming, and thus costly process. For the summarization itself, the hardest challenge is identifying what the most important concepts or knowledge in the text are.

Summarization can either be performed for a single document or a set of documents. The latter is called multi-document summarization, and adds new challenges to the summarization procedure. The identification of important concepts now needs to consider multiple sources, and concepts that are mentioned in multiple documents should not cause redundant information or an unnatural structure in the final summary [31].

2.6.1 Summarization Methods

As mentioned above, text summarization is a heavily researched area, and an abundance of summarization methods have been developed over the years. These methods have very different procedures and focus areas, and also different performance. The sections below present a subset of these methods which in our opinion illustrate the broad spectrum of approaches.

Frequency-Based Text Summarization

There are several variations of frequency-based summarization, but one of the most well known approaches is the SumBasic method [32]. SumBasic weighs sentences based on the word frequency distribution in the document, and these sentence weights are used to select relevant sentences to the summary. All words are assigned a probability score based on their number of occurrences in the document. Selecting sentences for the summary is an iterative process, where each iteration contains three steps. The first step is to find the word with the highest probability score, and the second is to pick the sentence with the highest score that contains this specific word. The final step is re-ranking, where the weights of each word are updated to ensure that too similar sentences are not added to the summary.

An experiment [32] carried out by the researchers shows that the re-ranking technique is very successful, and SumBasic achieves significantly higher ROUGE scores when the re-ranking is enabled. Without re-ranking, SumBasic is slightly outperformed by LexRank, which is described in the section below. With re-ranking, however, SumBasic achieved superior ROUGE score and outperforms LexRank by approximately 26%.

Graph-Based Text Summarization

LexRank [33] is one of the most commonly used graph-based methods, and tries to generate a summary by finding the most central documents in a document. The assumption is that these central sentences will cover the most important content of the document. To identify these sentences, LexRank constructs a graph consisting of nodes and edges, where the former represents sentences and the latter represents the similarity between sentences. To determine how central a sentence is, the number of edges connected to its associated node is counted. The higher the number of edges, the more central the sentence is.

Topic-Based Text Summarization

The main idea behind topic-based text summarization [34] is to identify dominant topics in texts, and to construct summaries reflecting the most important information related to these topics. Topic Model is an unsupervised method and processes unlabeled documents. In other words, the topics are generated based on the documents' content, and do not need labeled input to work.

Latent Dirichlet Allocation (LDA) [35] is a popular method for extracting important topics. LDA is an unsupervised probabilistic method that produces an a priori determined number of topics, where each topic is a cluster

with a fixed number of similar or related words. Each word in the cluster is associated with a topic probability, which defines how likely it is that this word was generated by the specific topic. An example of how a topic can be represented is illustrated in Figure 2.4. Unlike many other clustering methods, a word can belong to several clusters with various probabilities. LDA makes the *bag-of-words* assumption, which means that the order of the words in the document does not matter [34].

```
Topic: 1
Words: 0.022*"role" + 0.019*"develop" + 0.014*"play" +
       0.014*"gene" + 0.013*"diseas" + 0.011*"brain" +
       0.011*"cell" + 0.008*"express" + 0.007*"may" +
       0.007*"centrosom"
```

Figure 2.4: Example of how a topic is represented. The figure shows the probability of each word related to the specific topic.

Neural Network-Based Text Summarization

Using neural networks for summarization purposes is a quite new approach, and the first stepping stone was published by Kågebäck et. al in 2014 [36]. In general, neural network-based methods follow the same summarization procedure [37]. The first step is to construct word embeddings by translating words into continuous vectors. These vectors are then used to turn sentences into vectors. Finally, the sentence vectors are used as input to an extractive or abstractive summarization model. Neural networks can be used to either construct word embeddings for words or sentences, or for the selection or generation of sentences. One can use neural networks for all of the steps, or one can perform some of the steps with more traditional methods.

Generally, neural network-based models often perform better than traditional summarization methods [37]. Nevertheless, even though neural networks seem to revolutionize the summarization domain, there are still several challenges that remain to be solved. The most noteworthy challenge is that neural network-based models provide little or no transparency into how the summary is generated. Another challenge is that these models are not well suited for small datasets or long text sequences.

2.6.2 Evaluation

Several methods exist for evaluating how successful a constructed summary is. Some of these methods are presented in the sections below.

ROUGE

Recall Oriented Understudy for Gisting Evaluation (ROUGE) [38] is the most popular metric for evaluating summaries, and is a recall-based measure. ROUGE measures the overlap between the constructed summary and an associated gold standard summary, which is usually human-made. This overlap, also known as recall, is calculated by finding what ratio of words in the human-made summary had a match in the generated summary, and can be measured by using Equation 2.5 [38]. To produce human-made summaries is a costly process, and that is a disadvantage with using ROUGE. ROUGE calculates three evaluation metrics: precision, recall and F-measure [39].

$$ROUGE = \frac{Word\ matches\ between\ reference\ and\ hypothesis}{Length\ of\ reference\ summary} \tag{2.5}$$

The ROUGE scores can be computed for different degrees of overlap: ROUGE-L looks at the longest matching sequence, ROUGE-SU matches both skip-bigrams and unigrams, while ROUGE N-gram looks at n subsequent words. With ROUGE N-gram, n subsequent words must be the same in the reference summary and the generated summary to be considered a match. ROUGE N-grams with a certain n is usually named ROUGE-N, for instance ROUGE-1 matches unigrams and ROUGE-2 matches bigrams. When using unigrams to compute the ROUGE score, the word order in the constructed summary is irrelevant. With bigrams, on the other hand, one evaluates pairs of subsequent words. ROUGE-2 will not reward word occurrences unless they appear together and in the correct order. ROUGE-1 has shown to perform very well for evaluation of short summaries [38]. However, it is possible to achieve a high ROUGE score even though the summary is badly written, since ROUGE only measures word overlap and not how well-written the summary is [39]. Additionally, since ROUGE only considers word matches, it can cause false negatives when different spellings or variations of the same word are used.

BLEU

BLEU [40] is another metric for evaluating summaries, and it evaluates the textual quality of summaries that are translated by a machine from one natural language to another. The authors of the research emphasize that BLEU could be adapted to evaluate summarization algorithms, but it is not suited for this without modifications. However, Madnani, Tetreault, and Chodorow [41] found that BLEU performed well for paraphrase detection.

BLEU is one of the earlier evaluation metrics, and was part of the inspiration for creating ROUGE [38]. However, in contrast to ROUGE, BLEU focuses on precision rather than recall. The precision score is calculated by finding what ratio of words in the generated summary had a match in a reference summary, and can be measured by using Equation 2.6 [40].

$$BLEU = \frac{Word\ matches\ between\ reference\ and\ hypothesis}{Length\ of\ hypothesis\ summary} \tag{2.6}$$

The Human Reference Approach

The Human Reference Approach is an evaluation method that is time-consuming, but easy to execute [42]. This method involves a person verifying or comparing the constructed summary against a human-made summary. Since humans are subjective and may have biases, the results from such a test can only be used as a rough and inaccurate evaluation tool. This approach can be useful for supplementing results from other evaluations, rather than being the only evaluation performed.

Chapter 3

State of the Art

This chapter presents research related to our final search system, which is described in Section 4.2. First, related work is presented, and this section is divided into three different parts: ranking, clustering and summarization. Each of these parts is researched thoroughly, and there exists a plethora of relevant work. We have chosen a subset of research that we found especially relevant, based on either their approach or the results they achieved. We also strive to illustrate the breadth of approaches, while still keeping them relevant to our work. The second part of this chapter presents existing information retrieval systems within the medical domain, and explains why these are not adequate for our purpose and thus why there is a need for a new system.

3.1 Related Work

3.1.1 Ranking with Biomedical Texts

Ramampiaro and Li [43] developed a retrieval system called BioTracer, which is adapted for the biomedical domain. The underlying ranking models are Vector Space Model (VSM) and Okapi BM25, but BioTracer also uses other techniques to improve the models. One of the techniques that is used is to expand the query with other words with the same stem as a query term, and another is to allow boolean expressions like "cancer AND tumor". A user of the system can also choose which part of the document is important to them, for instance the abstract, or they can mark important terms in the query with a special syntax. Additionally, BioTracer supports interactivity by showing the user suggestions for queries while they are typing. Experiments showed that their extended models achieved significant improvements from the baseline ones.

Aravind et al. [44] proposed a retrieval system with Okapi BM25 as the underlying scoring function. Their approach consists of three different steps, where the first step is to index documents for more efficient document retrieval. Further, queries are expanded by using MetaMap[1], which increases the recall ability of the system. Finally, the ranking step is improved by re-ranking the documents, and this is performed by using Learning To Rank (LTR) algorithms. Experiments conducted by the authors showed that their extended approach had notably better results than the baseline model.

Wang, Zhang, and Yuan [45] presented a retrieval system with query expansion performed in two stages. First, the user inputs a query, receives a ranked document list and thereafter provides relevance feedback. This feedback is used to expand the user-originated query with terms that were common in the relevant documents. Further, a latent semantic relevance model identifies concepts that are relevant to the query by using tensor factorization and expands the query even further. Finally, the expanded query is used to "re-rank" the documents that were relevant. Experiments show that their approach achieves a noteworthy increase in performance from the baseline model, and according to the authors, their research can potentially be applied to multiple other fields, one example being recommender systems.

Xu et al. [46] introduced a supervised approach for retrieving biomedical documents. The main idea is to improve the retrieved results by using a trained query expansion model. This model is trained by using relevance feedback, and suggests expansion terms that make the results more relevant, but at the same time not too similar. Next, a new retrieval is initiated with the expanded query. The expansion process is performed by using three different optimization algorithms, where two of them are used to select terms to expand the query with during run time, and the last is a Learning To Rank algorithm used to improve the ranking of candidate terms. Their experiments show great promise, as their model outperforms the baseline models they tested against.

3.1.2 Clustering

Paulsen and Ramampiaro [47] introduced a hybrid clustering method for the biomedical domain. It combines the traditional K-means algorithm with Latent Semantic Indexing (LSI), where the latter is used to retrieve documents to cluster based on concepts. Using concepts means that documents do not need to exactly match the query to be retrieved, which results in a better basis for the clustering step. Additionally, K-means was modi-

[1] https://metamap.nlm.nih.gov/

fied to perform in two iterations, where the objective of the first iteration is to maximize the distance between centroids. This is performed by specifying a similarity threshold, and only adding documents to a cluster if their similarity is higher than this threshold. The centroid placements from the first iteration are then used as the starting cluster centroids for the second iteration. Experiments show that clusters become more distinct with their approach, and that the algorithm ends up being less greedy than baseline K-means.

Karaa et al. [48] proposed an algorithm for clustering MEDLINE abstracts, and their approach is based on a genetic algorithm. Because genetic algorithms do not handle textual input very well alone, it is combined with the Vector Space Model. The genetic algorithm is also dependent on good initial data for quickly finding the optimal result. To ensure that the initial data is of high quality, they use an agglomerative clustering algorithm. The authors found their approach to be quite efficient, and they claim that their solution is suitable for textual documents from any domain.

Yan et al. [49] proposed a clustering algorithm based on topic modeling. More specifically, they focus on applying topic modeling to short documents, which is a challenge because it is harder to identify conceptual patterns with a low volume of content. To handle this, they base their topic modeling on the corpus' "biterms", which the authors define as *unordered word-pairs co-occurred in a short context*. Basing the topic models on biterms is, according to the authors, an advantage since it is easier to generate topics this way than with a document based approach. Their experiments show that their combined approach not only performs well for short texts, it also achieves good performance for longer texts. Compared with baseline LDA, the proposed approach performs better in most cases.

Bui et al. [50] introduced a clustering algorithm based on Latent Dirichlet Allocation (LDA) and K-means. LDA and K-means are combined during clustering in order to utilize the topic information that LDA gathers. First, LDA is used to calculate a topic distribution for each document, and the result is then used as input to the K-means clustering. Since K-means clusters items based on distances, the authors had to experiment with different variants of distance metrics, and they found that *Probabilistic-Based Measurements* served the purpose better than *Vector-Based Measurements*. Their experiments also show that if clustering with the right amount of topics, their combined LDA and K-means approach performs notably better than the Vector Space Model and an approach where documents are assigned to clusters based on the topic they score highest on.

3.1.3 Summarization

Reeve, Han, and Brooks [51] combined a concept-based method with a frequency-based method, more specifically BioChain and FreqDist, respectively. Their idea was to identify important sentences based on concepts, but also use the frequency distribution of the document to decrease duplicate information by constructing a summary with a similar frequency distribution. Their experiment was conducted on medical texts and shows that the combined method, ChainFreq, achieves increased ROUGE-SU4 scores, and thus outperforms the individual models BioChain and FreqDist for this metric. However, for ROUGE-2 scores, ChainFreq is outperformed by FreqDist. In conclusion, their combined method performs well, but it does not achieve superior performance. It should be noted that this work is not as recent as other literature in this section, but we still find it highly relevant today.

Liu et al. [52] used Latent Dirichlet Allocation (LDA) as the underlying summarization method for identifying topics in the documents. To account for the fact that some topics might be less relevant, LDA topic weights are combined with word frequencies and the position and length of a sentence. Their combined method performs quite well compared to baseline models for LDA and word frequency (*SumBasic*). The methods were evaluated with several ROUGE variants, and the combined method achieves superior performance for all of them. One can therefore conclude that combining sentence characteristics with LDA can decrease the negative impact of irrelevant topics.

Gao et al. [53] presented a hybrid of graph-based summarization and topic modeling, and it uses LDA to generate topics. Further, these topics and the document sentences are represented by nodes in a graph, where edges between the topic nodes and sentence nodes represent an association for the topic-sentence pairs. The sentences are scored by calculating its associated edge weights using reinforcement learning, and these scores are used to select sentences to form a summary. Experiments show that using LDA to identify topics is quite successful and that their hybrid approach performs better than other similar methods.

Yin and Pei [54] proposed a neural network-based method called CNNLM, which uses convolutional neural networks (CNN) to construct a language model for the document sentences. The CNNLM identifies "features" of the different sentences, and uses the cosine similarity metric to calculate how similar sentences are. The selection of sentences is performed by using an optimization function, which calculates a score for each set of sentences, based on the redundancy between them and the importance of each sentence. Their experiments show that CNNLM performs quite well compared

to traditional methods such as LexRank, but it is also superior compared to other extractive neural network-based methods [37].

3.2 Existing Information Retrieval Systems

This section evaluates three different retrieval systems within the biomedical domain: PubMed, BioMedSearch and MedlineRanker. An overview of each system is described, in addition to including evaluations of the systems and why they are not adequate for our purpose. It should also be noted that Quertle[2] and MScanner[3] were considered for inclusion in this section, but since their approaches are comparable to BioMedSearch and MedlineRanker, respectively, we chose to exclude them for conciseness.

3.2.1 PubMed

PubMed [55], released by the National Center for Biotechnology Information in 1996, is the most extensively used search engine for the biomedical domain [56]. It consists of medical data from several sources, where the MEDLINE database [57] is the main source of data and makes the largest contribution with its over 25 million citations within the biomedical field. Other large sources of data for PubMed are PubMed Central[4] and Bookshelf[5], which provide references and citations from medical journals and online publications, respectively. In total, PubMed consists of over 30 million medical abstracts and citations. It should be noted, however, that PubMed itself does not provide the full content of articles, but instead provides abstracts and, if available, a link to the complete document.

PubMed uses Medical Subject Headings (MeSH) for indexing and retrieval of documents [56]. MeSH is a thesaurus developed by the National Library of Medicine, and it associates terms with concepts to make it easier to retrieve relevant documents without relying on exact query term matches. In addition to indexing, MeSH is also used for query expansion purposes. More specifically, the query is expanded with other relevant terms based on the MeSH concepts of the query terms. This is performed after transforming the query into a boolean form with "AND" between the terms in the query.

[2]https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3066589/
[3]https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-108
[4]https://www.ncbi.nlm.nih.gov/pmc/about/intro/
[5]https://www.ncbi.nlm.nih.gov/books/

Evaluation

There is no doubt that PubMed is a well-developed search engine for the medical domain. It offers several useful features, such as suggesting articles that may be related to the one being read [58]. Additionally, it handles the problem where in many cases, users strive to find the most recent research within a specific field. To solve this, PubMed offers to order the search results such that the newest articles appear first, allowing the user to get an overview of new research [56].

However, PubMed also has some significant disadvantages. The most prominent downside is the large amount of search results returned for each query, as shown by a PubMed log analysis [59]. The authors behind this analysis found that about a third of all queries returned more than one hundred results. To review these documents is a time-consuming task for the users of the system, and time is a precious resource for medical personnel. That this is a significant disadvantage is also supported by a senior researcher at the Department of Clinical and Molecular Medicine at NTNU. The researcher said that during a search, he would often need to narrow the search by adding more criteria, as there were too many results to go through with most queries. Further, the researcher points out that to use PubMed effectively, one needs to be experienced with the tool, as queries need to be very accurate to find the right results. This can be a challenge for users, especially as they may not always be able to figure out the right query input to find the documents they need.

In conclusion, PubMed is not a suitable tool to support the explainability in the In-Motion system. The disadvantages presented above are in conflict with our use scenario, as our goal is to provide decision support to medical experts who are searching for information related to a diagnosis and are often in a hurry.

3.2.2 BioMedSearch

BioMedSearch [60] is another search engine for retrieving medical texts, and it became commercialized in 2009. It includes all the data from PubMed and MEDLINE, in addition to research documents like theses, dissertations and more [61]. The system first retrieves relevant results and thereafter clusters them [61], allowing users to explore other articles within the topic they are reading about.

The search process in BioMedSearch is initiated by the user constructing a query [60]. The query terms are then mapped to their corresponding concepts, which are used to retrieve and rank documents that are presented to the user. The concepts are generated by using Unified Medical Language

System (ULMS)[6], which is based on several medical concept sources, for instance the MeSH concepts as described in 3.2.1. Further, the user can mark documents as relevant, and this relevance feedback is then input to an *association mining technique*. This algorithm analyzes the content in the relevant documents and improves the mapping between queries and concepts in the system. This results in an improved search result the next time the query is performed.

Evaluation

BioMedSearch has several advantages, for instance its clustering abilities [61]. Users are, as mentioned earlier, able to explore other articles within the same cluster, but they can also explore patterns in more depth by reviewing "sub-clusters", which means that the documents within a cluster were analyzed to find smaller clusters inside the cluster. Inspecting sub-clusters can be very helpful when the outer clusters are not specific enough and one needs to inspect narrower sets of documents. Furthermore, queries can be expressed both in a complex form, for example long questions, or a simple form like relevant keywords [60]. This differs from PubMed and many other systems that only support simple queries.

However, even though relevance feedback can improve search results significantly, BioMedSearch is dependent on such feedback to achieve high performance [60]. In their system evaluation, the MAP scores improve by between 40 and 56 percent from the first iteration to the second, indicating that feedback is critical for good system performance. This requirement can make it difficult to start using such a system, as results will be significantly worse in the beginning.

All things considered, BioMedSearch is not adequate for our purpose. As mentioned earlier, the medical personnel is dependent on quick and accurate search results that answer their questions, and given the fact that BioMedSearch needs several runs with feedback for performing well, the system is not a good fit for this use case. The idea behind relevance feedback is good and useful in general, but our users do not have the time to "train" and thereby increase the performance of BioMedSearch. On the positive side, the sub-clusters seem promising for our problem, and this would probably have been an appreciated feature.

3.2.3 MedlineRanker

MedlineRanker [62] is developed by Fontaine et al. to retrieve and rank data from the MEDLINE database, and was released in 2009. MedlineRanker

[6]https://www.nlm.nih.gov/research/umls/index.html

is based on a supervised approach, and its goal is to allow users without a thorough knowledge of a domain to still be able to find relevant documents from MEDLINE.

To initiate the retrieval process, the user uploads abstracts that are relevant to the topic they want to explore. These abstracts are referred to as the "training set", and are analyzed by MedlineRanker. Next, the training set is converted into a model, which is then used to retrieve and rank documents from the MEDLINE dataset. To achieve even better retrieval results, the user is allowed to define a "background set" and a "test set". The background set, usually defined as the whole MEDLINE database, is used when creating the model for the uploaded abstracts in order to determine which features make the abstracts unique. The test set is usually a subset of the MEDLINE database, and it defines which documents should be used when ranking documents. A smaller test set will make retrieval faster, and if a user knows which domain they are searching in, they may be able to reduce the test set significantly. For each run, MedlineRanker returns documents along with a number for how certain it is that these results are good.

Evaluation

The authors behind MedlineRanker present several strengths with their system. For instance, MedlineRanker can be used for retrieval within any topic in the MEDLINE database, as it has not been specialized for a certain topic or type of query. Furthermore, they also state that users do not need to know a domain very well to use the system, which allows for easier exploration of new domains. MedlineRanker achieves high retrieval performance and delivers the search result within a reasonable time, especially considering how flexible the system is.

MedlineRanker also has a few weaknesses. To retrieve relevant documents, the user must provide a sufficient amount of abstracts for MedlineRanker to identify its concept patterns. Their experiments show that this amount should be between 100 and 1000 abstracts, preferably even more, which can take a substantial amount of time for a user to gather. Furthermore, as users need to find this many relevant abstracts, they do need a certain level of knowledge within the topic. Finding many relevant documents from the large number of medical papers that exist is a heavy burden for a user, and they may need to use a different search system just for this purpose.

To summarize, MedlineRanker is not a suitable system for our users' kind of task. It is too cumbersome and time-consuming for medical personnel to search for relevant abstracts associated with the topic they are exploring. Additionally, the medical experts are often searching for something specific

within the CP domain, and to find many abstracts that reflect this level of granularity is nearly an impossible task. An alternative solution would be to use more general abstracts and then examine a large number of results in the search for what they are looking for, but that would also be too time-consuming for the medical personnel, as the key point is to serve them the information they need as fast as possible.

Chapter 4

Approach

This chapter presents a new search system, which was created during this thesis to solve our goal of supporting explainability in the In-Motion tool. The chapter starts by addressing biomedical-specific challenges for information retrieval. Next, the system is presented in its entirety, before describing the different steps in the search procedure. This section also presents our approach to text summarization, and contains a thorough description of a novel algorithm. Finally, we describe the implementation of the system.

4.1 Challenges with the Biomedical Domain

Medical texts introduce several new challenges to the information retrieval field that are not present for general texts. In this section, we will present some of the challenges that we find relevant to our work. We have based this content on the research that was done by Ramampiaro and Li [43] for the BioTracer system.

One of the main challenges with the biomedical field is that medical texts contain a large number of domain-specific terms, such as anatomical terminology and disease names. The prevalence of these medical terms causes the documents to differ significantly from general texts, for instance in terms of identifying keywords for document indexing. As most information retrieval models are created to work well with general texts rather than specialized fields, models that are used for medical purposes might require special adaptations to account for its terminology.

Furthermore, medical texts have a high term ambiguity, meaning that many words may mean different things depending on the context. One example of this is the disease AIDS and the verb aids, which are hard to distinguish if the casing and context of the words are not considered. If a system does

not address this challenge, it may result in a large number of false positive matches.

Another issue for information retrieval systems is that there is no standard terminology for the domain. When researchers make discoveries, they must assign them a name. The fact that there is no standardized scheme for naming discoveries, combined with the frequency of discoveries, leads to a large number of new terms that medical IR systems must account for. The volume and diversity in the structure of these new terms make this a challenging task to address.

The fact that an entity can have several term variants is another challenge for information retrieval. Many of the terms that are useful to index become highly infrequent due to this, often occurring only one or a handful of times within the corpus. This makes it harder to effectively index documents, as it is challenging to determine which infrequent terms are important. Algorithms based on inverse document frequency will increase the weight of rarely occurring words, but this is not necessarily enough when terms are highly infrequent. Another solution that can address part of this challenge is to use a thesaurus to group term variants and treat these as one single entity, which would boost their collective frequency. However, this depends on access to a thesaurus specialized for information retrieval, and this is usually not the case for thesauruses in the medical domain [63].

To summarize, there are many challenges in the medical domain that traditional information retrieval methods are not created to handle. It should also be noted that the medical domain is challenging not only for retrieval purposes, but for text mining in general, which also includes clustering and summarization. For this reason, traditional models might require certain adaptations for processing medical texts in an effective and accurate manner. However, these changes might come at the expense of the system's performance with general texts.

4.2 System Overview

To solve the aforementioned challenges and reach our goal, we propose a search system that can be combined with In-Motion to enable medical experts to quickly and easily verify machine-generated predictions. Our focus is on building a system to be used by physicians and medical experts who often work with diagnostication of cerebral palsy, and the most important requirements for this system are that it is simple to use and that it increases the efficiency of the diagnostication procedure. The architecture of our proposed system is based on the work in our specialization project [3], and is described in depth in this section. We start by providing an overview

of how the system works and how the three parts fit together, and proceed to describe each step in detail.

To facilitate decision support for In-Motion, our proposed solution first retrieves relevant articles to the user's query and then provides them a preview of the articles' content. To reduce redundant information, related documents are clustered together and form a unified summary of the cluster content. This procedure is illustrated in Figure 4.1.

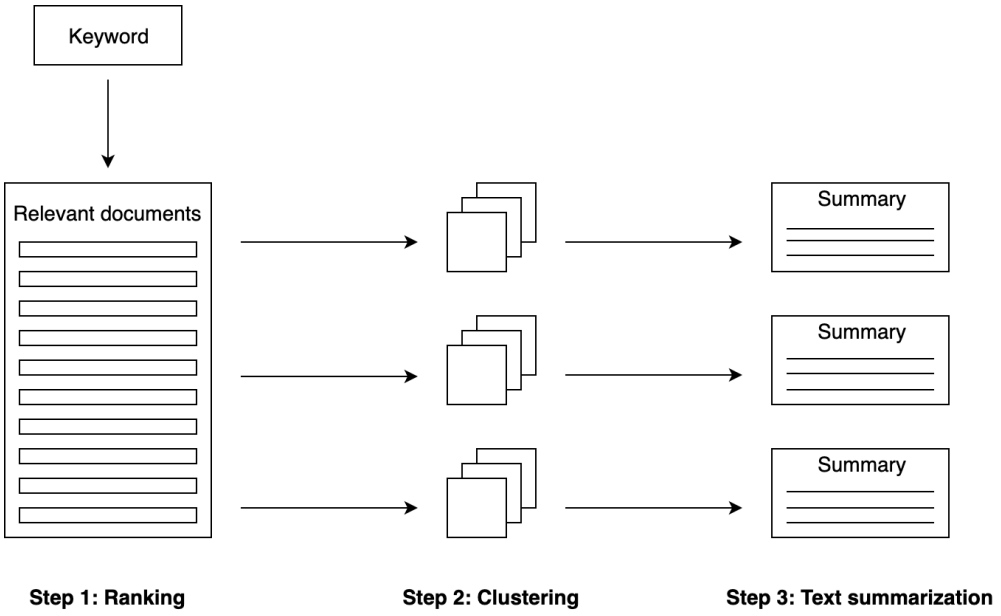

Figure 4.1: Illustration of the system procedure.

More specifically, the first step is to retrieve an a priori determined number of relevant documents for a given query. These documents are returned as a hitlist and ranked based on their relevance to the query. Further, the hitlist is grouped into clusters of related or similar documents to make it faster for a user to get an overview. Finally, a summarization algorithm analyzes the content of each cluster and extracts the most essential information into a summary that is relatively fast to digest.

The prerequisite for our system is that In-Motion explains the reasoning behind the prediction, or more specifically, outputs the prominent movement characteristics that lead to the given prediction. This is essential for medical experts to be able to understand and verify In-Motion's hypothesis. For the verification process to be as streamlined and user-friendly as possible, the long term plan is to integrate our search system with In-Motion, automatically using its explanatory characteristics as the query input in our system. However, this integration should not remove the ability of users to construct their own queries, as this might be necessary for experts to find

the information they need.

Another aspect of this system that we believe will be helpful to medical experts is that it can assist users with discovering new knowledge. As In-Motion analyzes movements of thousands of infants, it might discover correlations between specific movements and a high risk of cerebral palsy that the clinicians might be unaware of. Our search system can then play an important role in discovering knowledge about these features, allowing experts to extend their existing knowledge by learning from research that the system gathers. Considering the large amount of research that is published, this feature might become an important tool for medical personnel to discover new research.

Since our proposed solution is developed independently of In-Motion, our system is versatile and could be used for other medical purposes than diagnosis of cerebral palsy. As described above, searches can be initiated both by using the prominent movement characteristics from In-Motion or by the clinicians themselves. Our system can therefore potentially be used for other tasks than verifying predictions, for instance as a general search engine when researching other medical fields.

4.2.1 Ranking

The ranking step is the first step in our system, and it takes input from a user in the form of a list of keywords. The ranking then proceeds to find the 1000 most relevant documents, and the corpus for this retrieval will be the MEDLINE database. The most relevant documents are then returned, ordered by their relevance to the input keywords. As described earlier, there exists a lot of research in this area, but achieving high retrieval performance on medical texts remains a challenge.

TREC 2007 Genomics Track [64] was used for evaluating the retrieval and ranking performance, and it was chosen because there are no newer retrieval datasets that provide human judgements for evaluation. TREC is an annual text retrieval conference, and it provides a comprehensive dataset that can be used to evaluate ranking methods. The dataset for the TREC 2007 challenge contains passages extracted from around 160,000 documents, and the task is to identify and rank relevant passages for a given set of queries. These queries are formulated as questions about various medical topics, and the ability to retrieve relevant passages for these topics is one of the aspects that TREC 2007 measures. To evaluate the retrieval and ranking abilities of the methods, TREC provides *gold standards* associated with each query. The gold standards are produced by medical experts [65], and are considered the optimal rankings for the given queries. After the retrieved documents are ranked, the hitlist is compared to the gold stan-

dard. TREC evaluates how well submissions perform on both document, aspect and passage retrieval, and are reflected in the Document, Aspect and Passage2 scores, respectively.

Both Okapi BM25 and Language Model were considered as the underlying retrieval and ranking model. To assess their ranking capabilities, we evaluated both models with the TREC 2007 Genomics Track, and this evaluation is described in detail in Section 5.1. The MAP scores for Document, Aspect and Passage2 showed that Okapi BM25 performed 13.7%, 21.2% and 42.0% better than Language Model, respectively. Recall values were also calculated for these runs, and Okapi BM25 achieved a 6.0% higher score than Language Model. Our findings of Okapi BM25's performance are consistent with other research [43, 66, 67], and we therefore decided that Okapi BM25 was the most appropriate model for our needs.

4.2.2 Clustering

The second step of our approach is to cluster the documents that were retrieved in the first step. The purpose of this step is to group documents into clusters of related documents, which makes it easier for users to get an overview of the results and narrow their search. After the retrieval step, all documents should be fairly similar to the query, but the goal of this step is to group documents with finer granularity. For instance, when searching for CP-related documents, one cluster may be related to typical movements, while another may be related to postures. This step uses the set of relevant documents from the first step as input, and it outputs a list of document clusters, where each document is assigned to exactly one cluster.

Since the TREC 2007 documents are not labeled, we used unlabeled metrics to evaluate the separation of documents into clusters. As described in Section 2.5.2, silhouette score is a commonly used evaluation method for unlabeled data, and we therefore used this approach for evaluating our clustering implementation.

An important prerequisite for high clustering performance is determining the correct number of clusters (k). As mentioned in Section 2.5.1, this number must be determined a priori. The input size to this step will always be 1000 documents, and we therefore found it practical to identify a single value for k that contributes to a generally good clustering for a corpus input of this size. Determining the optimal k for each search would slow down the search significantly, and we therefore chose to use the same k for every query.

Both K-means and Topic Model were considered as the clustering method. To examine which method is best suited for our purpose, both methods

were tested with a subset of the queries and corresponding hitlists from the TREC 2007 dataset. We then found which number of topics or clusters gave the best scores. The number of topics for Topic Model was determined by calculating the silhouette scores for two topics and up to 30. The ideal number of topics is the one that gets the maximum silhouette score. Topic Model achieved the best results for only two topics, which implies an average size of 500 documents in each cluster. To determine the number of clusters in K-means, we used both the silhouette metric[1] and the elbow method[2]. We found that k-values between two and seven generally yielded the best results. To determine which method was better, we compared the best silhouette score of both methods for each query. We found that the K-means score on average was 27 times higher than the Topic Model score. In addition to this, we found it unlikely that two topics were enough to make distinct clusters in our use case. Based on these results, we chose K-means as the clustering algorithm for this project.

4.2.3 Summarization

The third and final step of our system is the summarization of clusters. In this step, a summarization algorithm analyzes the documents within each cluster and generates textual summaries containing the most essential information. This step takes the clusters from the second step as input and outputs a set of summaries, one for each cluster. The idea behind this step is to provide previews of the article content, and thereby limit the time spent searching for relevant information. The medical experts who are using this system are often in a hurry, and the summaries should therefore be as short as possible, but at the same time include relevant information. An average abstract is too long for this purpose, and we therefore aim to summarize the content with around 100 to 150 words.

As mentioned in Section 2.6, despite much research, no generally good approach to text summarization has been found. One of the challenges is that it is difficult to evaluate summaries without human judgements or gold standard summaries, and these are both very time-consuming to gather. Further, it is difficult to create a general algorithm that determines which sentences in a document should be picked for a summary, as the importance of a sentence is often highly subjective [31].

Because we are combining multiple documents into a single summary, we face some additional challenges. Documents within a cluster should be closely related, and may therefore contain a lot of duplicate information. The summary, however, should not contain duplicate information, but still

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score
[2] https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

reflect the breadth of content inside the cluster.

Possible Approaches

Topic Model is one of the approaches we considered for the summarization step. The Topic Model approach can use Latent Dirichlet Allocation (LDA), as described in Section 2.6.1, to construct a topic model for the whole corpus. Each topic consists of the associated keywords and their topic probabilities. The Topic Model approach calculates the similarity between the topic probability distribution of a document and the topic probability distributions of its sentences, and chooses the sentences that have the highest similarity scores. It should be noted that our problem is multi-document summarization, so instead of calculating the similarity between a document and its sentences, we calculate the similarity between a cluster and its sentences.

As described in Section 4.2.2, Topic Model did not perform very well for clustering the hitlist documents. The documents within the hitlist are relatively homogeneous since they are retrieved based on the same query, and this results in the generated topics being very similar. To be more specific, the topics usually contained the same associated words with relatively similar weighting for each word. Based on this finding, we hypothesized that Topic Model alone will not perform very good on the summarization step either, and should therefore be combined with something else to achieve good results.

Word Frequency is another approach we considered for the summarization step. Word Frequency selects sentences based on their word probabilities, meaning how common the words in a sentence are in the cluster. It does not take word relations into account, but naively ranks a sentence high if it contains commonly used words. This means that Word Frequency may rank some sentences higher than Topic Model, especially if a sentence uses common words but does not have the same topic distribution as the cluster.

Neural network-based methods were also considered for the summarization step, as they perform quite well when used for extractive summarization. However, most neural network-based methods are black-box approaches and are unable to explain the reasoning behind the sentence selection. As the main goal of our system is to achieve transparency and explainability in the In-Motion tool, we concluded that a method based on neural networks would not be ideal for our purpose.

Our Approach

Based on the difference in how sentences are selected with Word Frequency and Topic Model, we hypothesized that combining them might be promising. A combined approach could alleviate some of the disadvantages of each approach, and thus might perform better. We decided to implement this combined approach by scoring each sentence with both approaches, and then weigh these scores equally when deciding which sentences to include in a summary. We hypothesize that our approach can identify sentences that have a similar topic probability distribution to the cluster, while also catching some important sentences that use common words but have different topic probability distributions. Our approach for generating a summary for a given cluster consists of nine steps, and these are described in Figure 4.2.

Step 1 Construct an LDA topic model with all documents from the hitlist as the corpus.

Step 2 Calculate the topic probability distribution for the given cluster.

Step 3 For each sentence in the cluster:
- (a) Calculate its topic probability distribution.
- (b) Calculate the cosine similarity between the topic probability distributions of the sentence and the cluster.

Step 4 Construct a word frequency counter based on the sentences in the cluster.

Step 5 For each sentence in the cluster: sum the word frequencies.

Step 6 Normalize all word frequency sums by dividing by the max word frequency score. This causes the word frequency scores to be numbers between 0 and 1.

Step 7 Calculate the combined sentence score by giving the topic scores and the word frequency scores 50% weight each.

Step 8 Sort sentences by their combined score.

Step 9 While summary length is less than the threshold: pick the top sentence from the sorted list and add it to the summary.

Figure 4.2: A detailed description of our novel summarization algorithm. The input to this algorithm is the hitlist from the retrieval step and a cluster of documents, and the output is a textual summary of the cluster's contents.

To the best of our knowledge, this algorithm is novel in that it combines topic modeling with word frequencies to achieve better summaries than either of the methods alone. Both methods have been researched extensively [30], but to the best of our knowledge, a combination of the methods has not been tried previously.

Evaluation

The summarization algorithm is evaluated using scientific papers from ArXiv and PubMed, and these datasets[3] are provided by Cohan et al. in conjunction with their research on neural abstractive summarization models [68]. The articles in these datasets are used both to build a topic model and as a corpus for constructing summaries. The summarization is evaluated by summarizing the article content, and then comparing it to its associated human-made abstract. It should be noted that the abstract is excluded from the article content, so no sentences from the human-made abstract are known to the algorithm. The constructed summaries and the abstracts are compared using ROUGE-1 scores, which are an effective evaluation metric for short summaries, as described in Section 2.6.2. The ROUGE scores are calculated with a Python library[4], which returns a list of unigram scores: F-measure, precision and recall.

One shortcoming with this evaluation approach is that it evaluates single-document summarization, while our method summarizes several related documents from the same cluster. Hence, this evaluation approach will not reflect the multi-document performance. However, it is a good indicator of whether or not the combined algorithm is an improvement.

To determine the optimal number of topics, we computed ROUGE scores for summaries with different numbers of topics, and the goal was to find which number yielded the highest F-measure score. For the single-document evaluation approach, as described in the paragraphs above, 10 topics generally yielded the best results. For our system, however, we had no human-generated summaries to compare with, and could therefore not calculate any ROUGE scores.

4.3 Implementation

By combining the three steps outlined in Section 4.2, we can construct a complete search system. This system is described below and is illustrated in Figure 4.3.

[3] https://github.com/armancohan/long-summarization/
[4] https://github.com/pltrdy/rouge

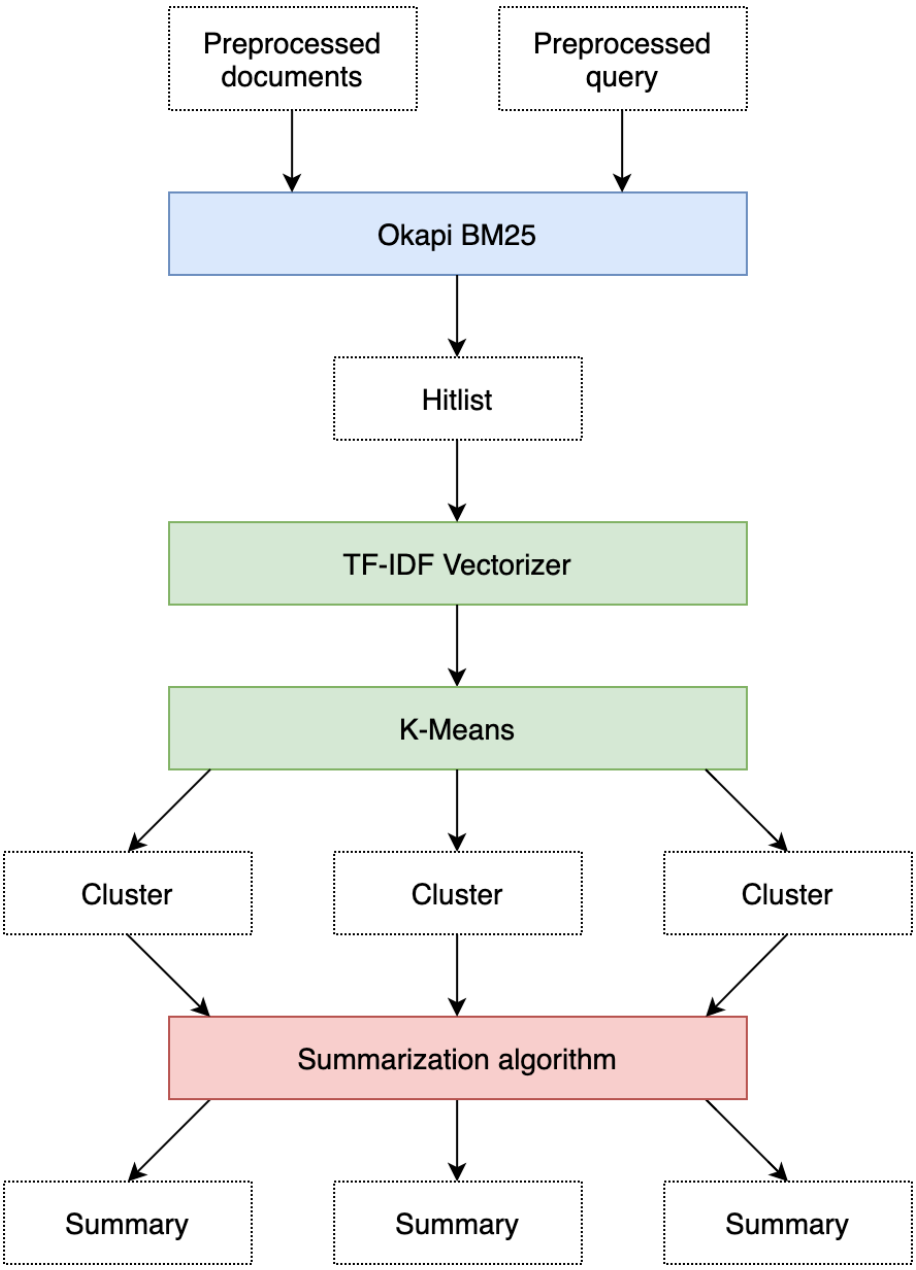

Figure 4.3: Illustration of the experiment implementation.

First, query-relevant documents are retrieved using Okapi BM25, which is implemented with a Python library called *tantivy*[5]. For initialization, tantivy constructs an index with the document corpus to enable quick retrieval of documents. Prior to the retrieval, the documents are cleaned by using the following preprocessing techniques:

[5]https://github.com/tantivy-search/tantivy

1. **Stemming.** We use the SnowballStemmer from the NLTK library[6], which is a stemmer that supports many different languages. SnowballStemmer is also the recommended stemmer for the English language [69].
2. **Stopwords removal.** This preprocessing step is also implemented with the NLTK library, and uses their built-in list of stopwords.
3. **Converting uppercase letters to lowercase letters.** This technique ensures that the casing of all words is the same, and was implemented with the built-in lowercase function in Python.
4. **Tokenization and character removal.** The documents were tokenized, meaning that they were divided into a list of words. Okapi BM25 operates on words, so this step is a prerequisite for the ranking algorithm. Additionally, HTML tags, citations and special characters were removed.

It should be noted that the preprocessing is not a part of tantivy, but is an additional step that we implemented to improve performance.

We also considered implementing a query expansion technique as part of the preprocessing, since TREC submissions with this technique scored on average 20% better on Aspect MAP and Passage2 MAP [70]. This technique was also presented as potential future work in our specialization project [3]. However, due to time constraints, we decided that improving our summarization approach would have a larger impact on our system than implementing this technique. We have listed query expansion in future work, as we believe it can have a positive impact on our system if implemented.

All the documents are preprocessed, but the original text is also stored as it is needed for some steps. For instance, the cleaned documents are used for retrieval and selection of sentences to the summary, while the uncleaned documents are used for the final summary construction. This is because the cleaned sentences are unreadable for the end-user, and the summaries must therefore contain the original sentences. To deal with this, our retrieval step indexes documents with two fields: *content* and *content clean*. The search process performs comparisons on the cleaned document content, but the result is a list of original documents.

As the next step in the process, tantivy retrieves the 1000 most query-relevant documents. This hitlist is then used as input to the K-means algorithm, which analyses the hitlist and divides it into a predetermined number of clusters (k). K-means is implemented with a Python library

[6]https://www.nltk.org/

called *sklearn*[7], which requires the hitlist content to be input in vector format. To convert documents into vectors, we used sklearn's *TfidfVectorizer*. TfidfVectorizer converts documents into vectors with d dimensions, where d is equal to the number of unique words in the corpus. Each dimension represents a word, and the value of the dimension shows the TF-IDF score of the word in the given document.

The final, and perhaps most vital step is the summarization of the content in each cluster. As mentioned earlier, we chose to combine Topic Model and Word Frequency in our summarization approach. The LDA topic model is implemented with the sklearn library, while we implemented the word frequency part of the algorithm ourselves.

[7]https://scikit-learn.org/stable/

Chapter 5

Results

This chapter presents the results of our experiments with the ranking, clustering and summarization steps. We present the results from each step in a separate section, and discuss our findings in Chapter 6. Ranking and summarization also contain a comparison with results from related research. However, we did not include such a comparison for our clustering results, as results from other research can not be compared directly. To compare clustering results, one needs to use the same input data. Our clustering step uses results from the ranking step as input, and it is therefore not fair to compare our clustering to research that uses different input.

5.1 Ranking

5.1.1 Experiment Results

We conducted an experiment to compare the retrieval and ranking capabilities of Okapi BM25 and Language Model. The goal of this experiment was to find the most suitable ranking model for medical texts. As described in Section 4.2.1, Okapi BM25 and Language Model were evaluated using the TREC 2007 corpus. The results of this experiment are listed in Table 5.1 and Table 5.2.

Table 5.1: Mean Average Precision (MAP) for Okapi BM25 and Language Model using the TREC 2007 dataset.

Method	Document MAP	Aspect MAP	Passage2 MAP
Okapi BM25	0.2302	0.1498	0.0592
Language Model	0.2024	0.1236	0.0417

Table 5.2: Precision and recall scores for Okapi BM25 and Language Model using the TREC 2007 dataset.

Method	Precision	Recall
Okapi BM25	0.0671	0.6060
Language Model	0.0635	0.5719

Okapi BM25 performs signficantly better for all evaluation metrics. It should be noted that high recall performance is most critical, since the clustering step needs as many relevant documents as possible, but does not take the ordering of documents into account. Okapi BM25 achieves a 6.0% better recall score and a 5.7% better precision score than Language Model, which means that Okapi BM25 has better retrieval capabilities than Language Model. However, the precision scores of both methods are very low because their hitlists consist of 1000 documents for every query, while the gold standard contains considerably fewer documents. In other words, even though the algorithms retrieve relevant content for the queries, they are "punished" for returning too many results. In addition to the precision and recall results, Okapi BM25 achieves 13.7%, 21.2% and 42.0% better MAP scores than Language Model for Document, Aspect and Passage2, respectively. These results show that BM25 is superior to Language Model for the experiments we have conducted.

5.1.2 Comparison with other approaches

In our specialization project [3], we evaluated our BM25 implementation with other TREC submissions. We compared our approach to the average results of submissions [70], in addition to NLMinter [71] and UniNE1 [72], which are two approaches that had published papers. NLMinter performed the best in all three categories, while UniNE1 was within the top five. The full results of this comparison are presented in Table 5.3, and it shows that Okapi BM25 performed 23.6%, 13.0% and 48.7% better than average scores on Document, Aspect and Passage2, respectively. However, the more specialized algorithms we compared with performed up to 42.8%, 75.6% and 93.9% better than our Okapi BM25 approach.

Table 5.3: Comparison with other TREC 2007 approaches.

Method	Document MAP	Aspect MAP	Passage2 MAP
NLMinter	0.3286	0.2631	0.1148
UniNE1	0.2777	0.2189	0.0988
Okapi BM25 (our implementation)	**0.2302**	**0.1498**	**0.0592**
Mean value for all runs	0.1862	0.1326	0.0398

5.2 Clustering

5.2.1 Experiment Results

To evaluate the clustering capabilities of K-means and Topic Model, we performed an experiment where we calculated the silhouette scores for different models and parameters. The goal was to find out which clustering model and parameters performed best for processing hitlists from the ranking step. K-means runs were performed from $k = 2$ and up to 50, while Topic Model was run with number of topics from 2 and up to 30. This procedure was repeated for a subset of TREC 2007 queries, and for each query, the maximum silhouette score and the associated number of clusters or topics were noted.

The silhouette scores for K-means were in the range from 0.213 to 0.886, with an average of 0.549. As a silhouette score of 1 means perfectly distinct clusters with strong connection within each cluster, K-means performs quite well in some cases, and good on average. The silhouette scores for Topic Model, on the other hand, were in the range from -0.011 to 0.051, with an average silhouette score of 0.020. In other words, K-means performs on average over 27 times better than Topic Model. The silhouette scores for Topic Model are close to zero, which means that the clusters probably are overlapping and that the documents could have been placed in other nearby clusters. The worst result for Topic Model is in fact negative, which indicates that some documents have been placed in the wrong cluster. However, this negative result is marginally different from zero.

5.3 Summarization

Our proposed summarization algorithm is evaluated with several experiments. First, we compare the summarization capabilities of our proposed algorithm to the baseline Topic Model and Word Frequency. These comparisons are made for several summary lengths, and for two different datasets, one for the medical domain and one for science in general. To evaluate the performance of our proposed algorithm more broadly, we compare it to three other summarization methods using the same datasets as mentioned above.

5.3.1 Experiment Results

The results of the comparison between our proposed approach, Topic Model and Word Frequency are presented in Table 5.4. Our combined approach is superior on all metrics for a summary length of 120 words, which is within the ideal summary length interval. The ROUGE F-measure of our proposed algorithm is 9.5% and 2.9% better compared to Topic Model and Word Frequency, respectively.

Table 5.4: ROUGE scores for summaries with approximately 120 words for the PubMed dataset. This table contains both the baseline models and our combined approach, where the latter is highlighted in bold.

Method	F-measure	Precision	Recall
Topic Model	0.3215	0.4600	0.2656
Word Frequency	0.3422	0.4840	0.2841
Topic Model with Word Frequency	**0.3520**	**0.4985**	**0.2919**

Figure 5.1 presents the ROUGE F-measure scores using the PubMed dataset for our combined approach, Topic Model and Word Frequency with several summary lengths. The graphs show that our combined approach is superior for all summary lengths. The ROUGE score is only 1.1% better than Word Frequency for a summary length just beneath 100 words, but this performance gap increases significantly after increasing the summary length. For summary lengths around 120, 140, 165, 190 and 215, our combined algorithm performs 2.8%, 3.2%, 4.5%, 3.1% and 4.0% better than Word Frequency. Topic Model performs worst out of all three methods for all summary lengths. For summary lengths of around 120 words, which is an appropriate summary length for our problem, our combined approach performs 9.5% better than baseline Topic Model.

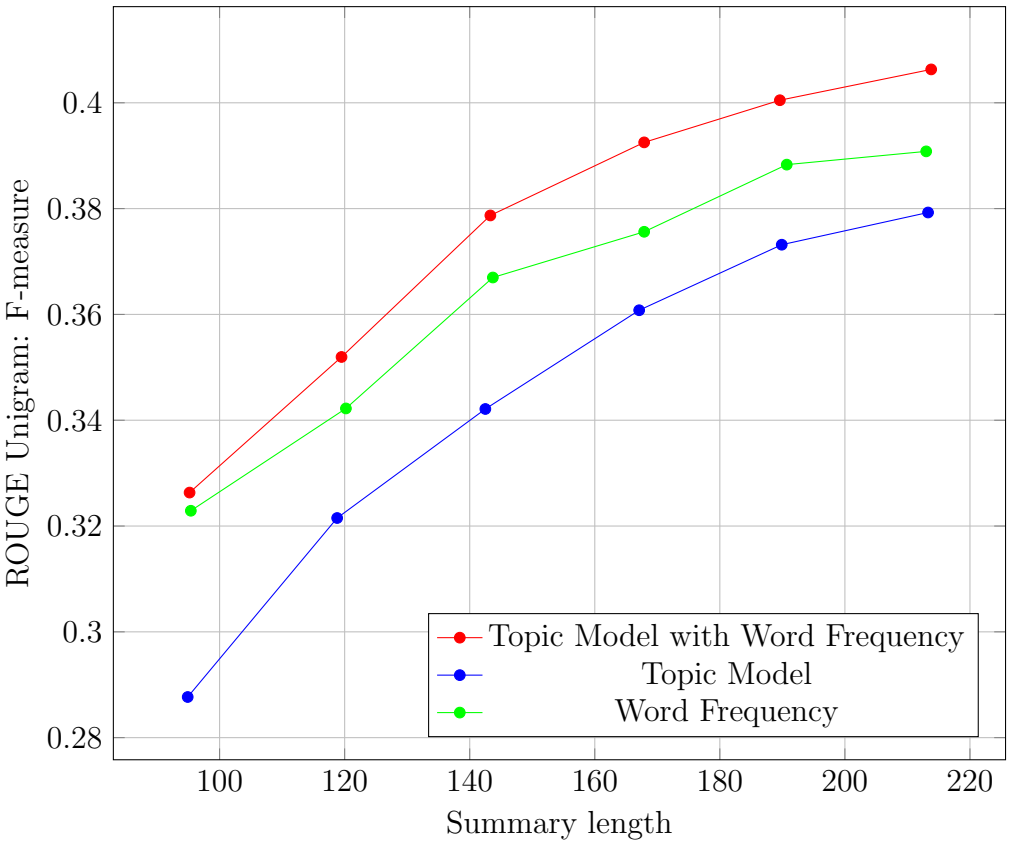

Figure 5.1: ROUGE scores for the PubMed dataset.

To assess the performance with another domain than the medical, the summarization algorithms were evaluated with another dataset consisting of scientific papers from ArXiv. The results from this run is visualized in Figure 5.2. It is clear that our algorithm does not only perform well with medical papers, but it performs remarkably well with scientific papers in general. The performance gap is significantly higher for the ArXiv dataset, and for a summary length of 120 words, our algorithm performs 9.6% and 12.7% better than Topic Model and Word Frequency, respectively.

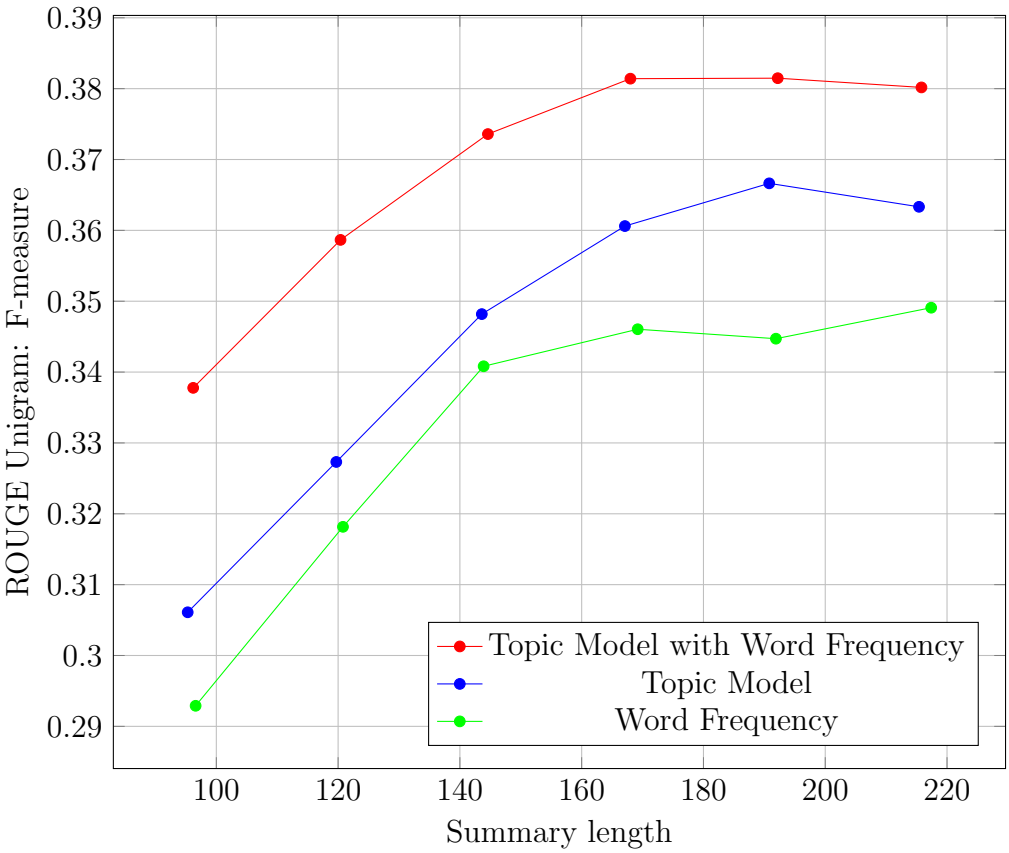

Figure 5.2: ROUGE scores for the ArXiv dataset.

5.3.2 Comparison with other approaches

The authors behind the summarization test corpus evaluated a few other extractive summarization methods with the same dataset [73]. These methods are SumBasic [74], LexRank [75] and Latent Semantic Analysis (LSA) [76]. SumBasic is a word frequency-based method, LexRank is a graph-based method and LSA, which was the inspiration of LDA [77] and is a topic-based method. The results for these methods are listed in Table 5.5. The summary length is on average 200 words for the PubMed dataset and 220 words for the ArXiv dataset.

Table 5.5: ROUGE scores for SumBasic, LexRank, LSA and our combined approach. All ROUGE scores are F-measures based on unigram matches, and the scores are listed for both the PubMed dataset and the ArXiv dataset. The average summary lengths for PubMed and ArXiv are 200 words and 220 words, respectively.

Method	PubMed F-measure	ArXiv F-measure
SumBasic	0.3715	0.2947
LexRank	0.3919	0.3385
LSA	0.3389	0.2991
Topic Model with Word Frequency	**0.4005**	**0.3802**

Our combined approach achieves higher ROUGE scores than the other approaches, even though the PubMed summaries produced by our approach are more concise by being on average about 10 words shorter. The same applies for the ArXiv dataset, where our approach achieves higher scores and is on average about 5 words shorter. Compared to the LexRank method which scored second on both datasets, our approach performed 2.2% and 12.3% better on the PubMed and ArXiv datasets, respectively.

Chapter 6

Discussion

This chapter discusses the results of our conducted experiments. We evaluate the three steps of our proposed solution, and reason about their performance results. As the ranking step is already evaluated thoroughly in our specialization project [3], the main focus of this chapter is the clustering and summarization steps, with an additional emphasis on the latter based on *RQ1*. However, we have included the most interesting ranking findings from our specialization project in Section 6.1, as they are relevant for research question *RQ2*.

6.1 Ranking

As described in our specialization project [3], our implementation of Okapi BM25 achieves quite high performance. It performs considerably better than the mean value of all TREC 2007 submissions, making our implementation competitive. There is still room for improvement as UniNE1 and NLMinter perform quite well compared to our Okapi BM25 approach, but we are confident that our Okapi BM25 implementation is a good stepping stone for further development.

During our specialization project [3], we also found that our preprocessing techniques were quite successful. Using preprocessing with Okapi BM25 gave a boost of over 59% across all three MAP scores compared to without preprocessing. The most interesting finding from the specialization project is that UniNE1 is similar to our implementation, but in addition to using the same preprocessing techniques, they also use query expansion. The fact that UniNE scored much higher than our approach indicates that implementing query expansion had a large effect on their performance.

6.2 Clustering

As presented in Section 5.2, K-means' silhouette scores vary a lot depending on the input data. The maximum silhouette score is approximately 4.2 times larger than the minimum silhouette score, which is a significant difference. It should be further investigated whether this is a query-specific issue, or if it is an underlying problem with the K-means algorithm. If the latter is the case, a potential solution might exist. Otherwise, the queries and their associated hitlists may not have any reasonable clusters. We have investigated the effect of input data on the silhouette scores of K-means and Topic Model, and found no clear evidence that some queries are unfit for clustering. The same query may result in an increased silhouette score for K-means, and at the same time a decreased silhouette score for Topic Model. In other words, there is no strong correlation between the results of Topic Model and K-means, which indicates that the chosen model is more important for the result than the input data.

Another interesting finding is the fact that K-means achieves remarkably higher silhouette scores than Topic Model, even though they have approximately the same number of clusters. We hypothesize that Topic Model performs poorly since it is generating topics based on a somewhat homogeneous hitlist. If the ranking step achieves good results, the documents in the hitlist should all be relevant to the query, and therefore contain a lot of the same information. We manually investigated the generated topics for several runs, and we found that the topics often were quite similar. The words associated with each topic were often the same, and the weighting for each word was also relatively similar. This could result in overlapping and indistinct clusters, where many documents hypothetically could be placed in more than one cluster. This was also confirmed by looking at the topic probability distribution of different sentences, which was often spread evenly between multiple topics rather than clearly belonging to a single topic. K-means, on the other hand, looks at the similarity of word occurrences rather than topic probability distributions to determine which cluster a document belongs to, and thereby compares with a finer granularity than Topic Model. As a result, K-means may be better at separating a somewhat homogeneous dataset. We hypothesize that Topic Model would perform much better if it were not for the ranking step.

6.3 Summarization

As mentioned earlier, our proposed summarization algorithm performs better than the baseline Topic Model and Word Frequency for both the PubMed dataset and the ArXiv dataset. This is an excellent result, as

it implies that our algorithm achieves high performance for both medical texts and scientific papers in general. An interesting finding is that the performance gap between our combined approach and the other two methods is significantly larger for ArXiv than for PubMed. This implies that our algorithm is an even larger improvement from the baseline models for scientific papers than for medical papers. Even though our focus area is the medical domain, we welcome these results as they show that the proposed summarization algorithm works well in more than one specialized scenario. It should also be noted that our algorithm is superior for all summary lengths, which means that our algorithm performs best in all possible cases. With this in mind, we are confident that our proposed algorithm is successful and that it should be used as the underlying summarization algorithm.

Another interesting finding is that Word Frequency outperforms Topic Model for medical articles, while the opposite is true for scientific papers in general. However, we struggle to find a reasonable explanation for the increased performance of Word Frequency with the PubMed dataset. Word Frequency constructs a frequency counter for each document, and should therefore be adaptive and not favorize medical texts over scientific papers in general. One potential explanation is that the medical summaries might be written in a similar fashion, which by chance happened to be a suitable structure for Word Frequency. As mentioned above, Topic Model performs worse than Word Frequency with the PubMed dataset, but not with the ArXiv dataset. One probable explanation for this is that Topic Model might generate better topics for several industries combined in one dataset than for one industry alone, for instance medicine. This could make sense since the Topic Model corpus would consist of a more heterogeneous dataset than if all documents are from the same industry. The performance of Topic Model with homogeneous datasets is discussed above in a clustering context, and the same reasoning can be applied here, as the topics used for summarization are clusters of words. Another hypothesis is that Topic Model coincidentally might be poor at generating topics for the medicine industry.

As described previously, the combination of Topic Model and Word Frequency performs significantly better than the two baseline models. We have several hypotheses as to why this is the case. First, as already described, we hypothesize that Topic Model is unable to create distinct and well formed topics because of the homogeneous corpus, causing the selection of sentences to be suboptimal. Further, Word Frequency is a quite simple and naive approach, and selecting sentences based on term occurrences alone might cause irrelevant sentences to be chosen for the summary. Even though the sentence selection is performed based on a cleaned dataset

without stopwords, irrelevant medical terms such as verbs might not be filtered out during the cleaning stage. We believe that our proposed algorithm utilizes the advantages of both approaches, or that the combination reduces the impact of shortcomings that each baseline approach has. For instance, Word Frequency might reduce the score of a sentence that was scored highly by Topic Model, based on the fact that it has many infrequent terms, which indicates that it may not be very important. The combination could also be advantageous the other way around, where Word Frequency originally would punish a sentence because of low term occurrences, but Topic Model might increase the score of the sentence because it contains terms that are important to the cluster's topics. This reasoning shows how the combination of Word Frequency and Topic Model can have a positive effect, as they can work together to give a more nuanced score.

As with other summarization approaches, our proposed algorithm produces both successful summaries and less successful ones. Figure 6.1 presents how a well constructed summary might look like, and it also contains the reference summary to illustrate how similar they are. This constructed summary achieved a ROUGE F-measure score of 0.654, which is quite good. One can see that several sentences in the constructed summary are identical to sentences in the gold standard summary. This example shows that our algorithm is capable of capturing important information from the original texts. However, our proposed algorithm is not always this successful, as illustrated in Figure 6.2. This constructed summary achieved a ROUGE F-measure score of only 0.156, which is not very pleasing. One can see that the algorithm misunderstands which sentences provide value to the constructed summary, as the algorithm chooses three almost identical sentences where only the numbers are changed. In this case, the algorithm would benefit from a technique that reduces redundancy in the summary, for instance by not selecting sentences if they are too similar to already picked sentences. This could easily be implemented using a distance measure like the cosine similarity, and the potential benefits from such a feature should be investigated further.

```
REFERENCE SUMMARY
purposeto demonstrate the usefulness of enhanced depth
imaging optical coherence tomography ( edi - oct ) in
investigating choroidal lesions inaccessible to
ultrasound sonography.methodsin a 60-year - old woman
with an asymptomatic choroidal nevus , normal oct was
used to observe the macula and edi - oct to image the
choroidal nevus that was inaccessible to ultrasound .
the exact location of the lesion in the choroid and
the dimensions of the nevus were measured.resultsthe
lesion was located in the superior macula , and the
nevus was homogeneous in its reflectivity .   we
observed a thickened choroid delineated by the shadow
cone behind it , measuring 1,376  325 m in the larger
vertical cut and 1,220  325 m in the larger horizontal
cut in an image with a 1:1 pixel mapping and automatic
zoom .   the macular profile and thickness were both
normal.conclusionsedi-oct appears to be an excellent
technique for measuring choroidal nevi and all
choroidal lesions accessible to oct imaging by
depicting their exact location in the choroid , their
dimensions , and their demarcation from the surrounding
healthy tissue , thus allowing for a more efficient
and accurate follow - up .

GENERATED SUMMARY
edi - oct appears to be an excellent technique for
measuring choroidal nevi and all choroidal lesions
accessible to oct imaging by depicting their exact
location in the choroid , their dimensions , and their
demarcation from the surrounding healthy tissue , thus
allowing for a more efficient and accurate follow - up .
in a 60-year - old woman with an asymptomatic choroidal
nevus , normal oct was used to observe the macula and
edi - oct to image the choroidal nevus that was
inaccessible to ultrasound .  recently , enhanced depth
imaging oct ( edi - oct ) has been used in order to
measure choroidal thickness in normal and pathologic
eyes [ 3 , 4 ] .
```

Figure 6.1: An example of a generated summary and its associated reference summary. This system-made summary is an example of a well constructed summary, and it achieved a ROUGE F-measure score of 0.654.

```
REFERENCE SUMMARY
indium - promoted coupling reactions between propargyl
aldehydes ( 1 ) and -chloropropargylphenyl sulfide are
reported .   although water has been shown to accelerate
indium metal promoted reactions , the reverse pattern was
observed in this series .   use of n - methylformamide
( nmf ) , which has not previously been a solvent known
for use in indium - promoted reactions , afforded an
acceleration of these barbier - style reactions compared
to water .   indium - promoted reactions in this study
also showed excellent regiocontrol and good stereocontrol
, allowing for easy entry into the formation of
epoxydiyne and enediyne skeletal structures .   this
paper also describes use of the barbier coupled product
( 2 ) as a new , and easy , entry into the formation of
enediyne and epoxydiyne skeletal structures .

GENERATED SUMMARY
separation was accomplished using radial chromatography
on silica gel ( 35:1 hexanesethyl acetate ) allowing
isolation of two hydroxyl sulfide diastereomers in a
75:25 syn / anti ratio : yield = total = 458 mg ( 1.
separation was accomplished using radial chromatography
on silica gel ( 35:1 hexanesethyl acetate ) allowing
isolation of two hydroxyl sulfide diastereomers in a
20:80 syn / anti ratio : total yield = 489 mg ( 1.
separation was accomplished using radial chromatography
on silica gel ( 35:1 hexanesethyl acetate ) allowing
isolation of two hydroxyl sulfide diastereomers in a
60:40 syn / anti ratio : yield = total = 434 mg ( 1.
```

Figure 6.2: An example of a generated summary and its associated reference summary. This system-made summary is an example of a poorly constructed summary, and it achieved a ROUGE F-measure score of 0.156.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has discussed the importance of transparency in AI systems, especially when used for clinical purposes such as diagnostication of cerebral palsy. We have shown that existing medical search engines are not sufficient for this purpose, and that there is a need for a new system. To address this, we proposed a system that provides features that clinicians need to efficiently verify machine-generated predictions. Because this system will be used for explainability in the medical domain, we built each part of the system to work well with medical documents.

Our study has confirmed earlier findings that when using Okapi BM25 as a similarity model, our system achieves good retrieval results for medical datasets. By combining this algorithm with several preprocessing techniques, our system was able to achieve high retrieval performance. With more time, we would also have explored the improvements that query expansion as a preprocessing technique could have yielded. We hypothesize that query expansion could be especially useful due to the many term variants within the medical terminology, and could therefore create a better basis for the clustering step. Our clustering approach also worked well for the medical domain, and we chose to use K-means over topic modeling because it was able to create more distinct clusters, even with the homogeneous input containing medical terminology.

We have also shown that our proposed summarization algorithm was successful in extracting the important information in the documents, and thus in creating useful summaries. Our sentence selection algorithm is based on equally weighing topic probability distribution and word frequency distribution, and is to the best of our knowledge a novel approach. Thus, this

combination of methods is our contribution to the summarization field. In addition to outperforming baseline Topic Model and Word Frequency for medical texts, the performance gap was even larger for scientific papers in general. As the focus area of this thesis has been the medical domain, this was a very interesting finding, and it speaks to the general utility of our novel summarization approach.

In conclusion, this thesis has successfully answered both of our research questions. Our proposed solution showed great potential for validating machine learning-based predictions, and summarization was the most important contribution to doing this effectively. The research conducted during this thesis is a valuable contribution to the In-Motion project, and can bring the project one step closer to being used extensively for diagnostication of cerebral palsy.

7.2 Future Work

As mentioned in the previous section, the exploration of query expansion is left for further work. We have faith that query expansion as a preprocessing technique will significantly improve the retrieval capabilities of our system, and thus improve its overall performance.

Furthermore, the summarization algorithm could be tuned and enhanced even further. Manually inspecting the summaries produced by our algorithm showed that there is room for improvement regarding redundancy. This could be solved by implementing a redundancy checker that only selects sentences to the summary if they provide new information, meaning that they are not too similar to the already existing content.

Another area that requires work is creating a user interface where users can easily explore clusters and their associated summaries. Clinicians should be heavily involved in this process, as their experience of the system is essential for it to be used extensively in their diagnosis procedures. Once the user interface is implemented and is actively used by clinicians, the algorithms for retrieval and summarization of documents could be improved in an iterative fashion. Furthermore, the system could increase its utility by collecting user-relevant feedback to improve itself automatically.

The most important task for future work is integration with the In-Motion system. As described earlier, our system is dependent on In-Motion outputting the movement characteristics that were prominent for a given prediction. Once this is implemented, our search system can be modified to automatically input these characteristics as query keywords, making the verification process seamless for the medical experts using In-Motion.

However, we do not intend to remove the clinicians' ability to formulate their own search queries, as this feature can be essential for finding the right material. Our search system may also need to be tuned to process and handle the characteristics input from In-Motion properly, depending on how they are represented by In-Motion. Overall, seamlessly integrating our search engine with the In-Motion system will help clinicians diagnose patients more efficiently and confidently.

References

[1] Oslo universitetssykehus. *Cerebral parese (CP) hos barn og ungdom.* 2017. URL: https://helsenorge.no/sykdom/hjerne-og-nerver/cerebral-parese-barn-og-ungdom (visited on 12/06/2019).

[2] Bryce Goodman and Seth Flaxman. "European Union regulations on algorithmic decision-making and a "right to explanation"". In: *AI Magazine* 38.3 (2017), pp. 50–57.

[3] Marie Kjellstrøm Thorkildsen. *How to use text information to support explainability in machine learning systems?* Project report in TDT4506. Department of Computer Science, NTNU – Norwegian University of Science and Technology, Dec. 2019.

[4] Norsk Helseinformatikk AS. *Cerebral parese (CP).* 2018. URL: https://nhi.no/sykdommer/barn/nervesystemet/cerebral-parese/?page=1 (visited on 12/06/2019).

[5] Christa Einspieler et al. "Cerebral Palsy: Early Markers of Clinical Phenotype and Functional Outcome". In: *Journal of clinical medicine* 8.10 (2019), p. 1616.

[6] NTNU Technology Transfer AS. *Lowering age of Cerebral Palsy detection and intervention.* URL: https://www.ntnutto.no/prosjekter-items/in-motion/ (visited on 12/06/2019).

[7] Christa Einspieler and Heinz FR Prechtl. "Prechtl's assessment of general movements: a diagnostic tool for the functional assessment of the young nervous system". In: *Mental retardation and developmental disabilities research reviews* 11.1 (2005), pp. 61–67.

[8] Siri Osland et al. *Spedbarns spontanbevegelsar: Utvikling av In-Motion-appen.* 2017. URL: https://fysioterapeuten.no/Fag-og-vitenskap/Blikk-paa-forskning/Spedbarns-spontanbevegelsar-Utvikling-av-In-Motion-appen (visited on 12/06/2019).

[9] Lars Adde et al. "General movement assessment: predicting cerebral palsy in clinical practise". In: *Early human development* 83.1 (2007), pp. 13–18.

[10] Mijna Hadders-Algra. "General movements: a window for early identification of children at high risk for developmental disorders". In: *The Journal of pediatrics* 145.2 (2004), S12–S18.

[11] Norwegian University of Science and Technology (NTNU). *The In-Motion app and testing*. 2019. URL: https://www.ntnu.edu/ikom/in-motion (visited on 12/06/2019).

[12] Daniel Groos and Kristian Aurlien. "Infant Body Part Tracking in Videos Using Deep Learning-Facilitating Early Detection of Cerebral Palsy". Master's thesis. Norwegian University of Science and Technology, 2018.

[13] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/. 2019.

[14] Defense Advanced Research Projects Agency. "Explainable Artificial Intelligence (XAI)". In: (2016). DOI: https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf.

[15] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models". In: *arXiv preprint arXiv:1708.08296* (2017).

[16] Bahador Khaleghi. *The How of Explainable AI: Post-modelling Explainability*. 2019. URL: https://towardsdatascience.com/the-how-of-explainable-ai-post-modelling-explainability-8b4cbc7adf5f (visited on 12/05/2019).

[17] Erico Tjoa and Cuntai Guan. "A survey on explainable artificial intelligence (XAI): towards medical XAI". In: *arXiv preprint arXiv:1907.07374* (2019).

[18] Andreas Holzinger et al. "What do we need to build explainable AI systems for the medical domain?" In: *arXiv preprint arXiv:1712.09923* (2017).

[19] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?: Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. ACM. 2016, pp. 1135–1144.

[20] Himabindu Lakkaraju et al. "Interpretable & explorable approximations of black box models". In: *arXiv preprint arXiv:1707.01154* (2017).

[21] Bing Liu. *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[22] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge university press, 2008, pp. 22–34, 219–235, 349–374.

[23] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009.

[24] Anna Huang. "Similarity measures for text document clustering". In: *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*. Vol. 4. 2008, pp. 9–56.

[25] Trupti M Kodinariya and Prashant R Makwana. "Review on determining number of Cluster in K-Means Clustering". In: *International Journal* 1.6 (2013), pp. 90–95.

[26] Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. "Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA". In: *Information Retrieval* 14.2 (2011), pp. 178–203.

[27] Jonathan Baarsch and M Emre Celebi. "Investigation of internal validity measures for K-means clustering". In: *Proceedings of the international multiconference of engineers and computer scientists*. Vol. 1. sn. 2012, pp. 14–16.

[28] Rosa Lletı et al. "Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes". In: *Analytica Chimica Acta* 515.1 (2004), pp. 87–100.

[29] Tippaya Thinsungnoena et al. "The clustering validity with silhouette and sum of squared errors". In: *learning* 3.7 (2015).

[30] Mehdi Allahyari et al. "Text summarization techniques: a brief survey". In: *arXiv preprint arXiv:1707.02268* (2017).

[31] Dipanjan Das and André Martins. "A Survey on Automatic Text Summarization". In: (2007). DOI: `http://www.cs.cmu.edu/~nasmith/LS2/das-martins.07.pdf`.

[32] Ani Nenkova and Lucy Vanderwende. "The impact of frequency on summarization". In: *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005* 101 (2005).

[33] Günes Erkan and Dragomir R Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization". In: *Journal of artificial intelligence research* 22 (2004), pp. 457–479.

[34] David M Blei. "Probabilistic topic models". In: *Communications of the ACM* 55.4 (2012), pp. 77–84.

[35] David M Blei, Andrew Y Ng, and Michael I Jordan. “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

[36] Mikael Kågebäck et al. “Extractive summarization using continuous vector space models”. In: *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*. 2014, pp. 31–39.

[37] Yue Dong. “A survey on neural network-based summarization methods”. In: *arXiv preprint arXiv:1804.04589* (2018).

[38] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[39] Yvette Graham. “Re-evaluating automatic summarization with BLEU and 192 shades of ROUGE”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015, pp. 128–137.

[40] Kishore Papineni et al. “BLEU: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics. 2002, pp. 311–318.

[41] Nitin Madnani, Joel Tetreault, and Martin Chodorow. “Re-examining machine translation metrics for paraphrase identification”. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics. 2012, pp. 182–190.

[42] K Sparck Jones et al. “Automatic summarizing: factors and directions”. In: *Advances in automatic text summarization*. 1. MIT press Cambridge, Mass, USA, 1999, pp. 1–12.

[43] Heri Ramampiaro and Chen Li. “Supporting biomedical information retrieval: The biotracer approach”. In: *Transactions on large-scale data-and knowledge-centered systems IV*. Springer, 2011, pp. 73–94.

[44] Manu Aravind et al. “A Modified Medical Information Retrieval System”. In: *2019 IEEE 9th International Conference on Advanced Computing (IACC)*. IEEE. 2019, pp. 218–222.

[45] Haolin Wang, Qingpeng Zhang, and Jiahu Yuan. “Semantically enhanced medical information retrieval system: a tensor factorization based approach”. In: *IEEE Access* 5 (2017), pp. 7584–7593.

[46] Bo Xu et al. “A supervised term ranking model for diversity enhanced biomedical information retrieval”. In: *BMC bioinformatics* 20.16 (2019), pp. 1–11.

[47] Jon Rune Paulsen and Heri Ramampiaro. “Combining latent semantic indexing and clustering to retrieve and cluster biomedical information: A 2-step approach”. In: *NIK-2009 conference*. 2009.

[48] Wahiba Ben Abdessalem Karaa et al. “Medline text mining: an enhancement genetic algorithm based approach for document clustering”. In: *Applications of Intelligent Optimization in Biology and Medicine*. Springer, 2016, pp. 267–287.

[49] Xiaohui Yan et al. “A biterm topic model for short texts”. In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 1445–1456.

[50] Quang Vu Bui et al. “Combining Latent Dirichlet Allocation and K-means for documents clustering: effect of probabilistic based distance measures”. In: *Asian Conference on Intelligent Information and Database Systems*. Springer. 2017, pp. 248–257.

[51] Lawrence H Reeve, Hyoil Han, and Ari D Brooks. “The use of domain-specific concepts in biomedical text summarization”. In: *Information Processing & Management* 43.6 (2007), pp. 1765–1776.

[52] Na Liu et al. “Topic-sensitive multi-document summarization algorithm”. In: *2014 Sixth International Symposium on Parallel Architectures, Algorithms and Programming*. IEEE. 2014, pp. 69–74.

[53] Dehong Gao et al. “Lda-based topic formation and topic-sentence reinforcement for graph-based multi-document summarization”. In: *Asia Information Retrieval Symposium*. Springer. 2012, pp. 376–385.

[54] Wenpeng Yin and Yulong Pei. “Optimizing sentence modeling and selection for document summarization”. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 2015.

[55] National Center for Biotechnology Information. *PubMed Overview*. URL: `https://pubmed.ncbi.nlm.nih.gov/about/` (visited on 05/21/2020).

[56] Zhiyong Lu. “PubMed and beyond: a survey of web tools for searching biomedical literature”. In: *Database* 2011 (2011).

[57] U.S. National Library of Medicine. *MEDLINE®: Description of the Database*. 2019. URL: `https://www.nlm.nih.gov/bsd/medline.html` (visited on 11/09/2019).

[58] Wei Wei et al. “Finding related publications: extending the set of terms used to assess article similarity”. In: *AMIA Summits on Translational Science Proceedings* 2016 (2016), p. 225.

[59] Rezarta Islamaj Dogan et al. "Understanding PubMed® user search behavior through log analysis". In: *Database* 2009 (2009).

[60] Yanqing Ji et al. "Integrating unified medical language system and association mining techniques into relevance feedback for biomedical literature search". In: *BMC bioinformatics* 17.9 (2016), p. 264.

[61] *BioMedSearch Home Page*. URL: `http://www.biomedsearch.com/` (visited on 05/21/2020).

[62] Jean-Fred Fontaine et al. "MedlineRanker: flexible ranking of biomedical literature". In: *Nucleic acids research* 37.suppl_2 (2009), W141–W146.

[63] William Hersh. *Information Retrieval: A Health and Biomedical Perspective*. 3rd. Health Informatics. Springer, 2008, p. 319. ISBN: 038778702X,9780387787022.

[64] *TREC 2007 Genomics Track Protocol*. 2008. URL: `https://dmice.ohsu.edu/trec-gen/2007protocol.html` (visited on 11/14/2019).

[65] Casey Lynnette Overby, Peter Tarczy-Hornoch, and Dina Demner-Fushman. "The potential for automated question answering in the context of genomic medicine: an assessment of existing resources and properties of answers". In: *BMC bioinformatics*. Vol. 10. 9. BioMed Central. 2009, S8.

[66] Xuheng Xu et al. "A comparison of local analysis, global analysis and ontology-based query expansion strategies for bio-medical literature search". In: *2006 IEEE International Conference on Systems, Man and Cybernetics*. Vol. 4. IEEE. 2006, pp. 3441–3446.

[67] Yanjun Li, Ningtao Shi, and D Frank Hsu. "Fusion analysis of information retrieval models on biomedical collections". In: *14th International Conference on Information Fusion*. IEEE. 2011, pp. 1–8.

[68] Arman Cohan et al. "A discourse-aware attention model for abstractive summarization of long documents". In: *arXiv preprint arXiv:1804.05685* (2018).

[69] NLTK Project. *Stemmers*. URL: `http://www.nltk.org/howto/stem.html` (visited on 11/28/2019).

[70] L. Ruslen W. Hersh A. Cohen and P. Roberts. "TREC 2007 Genomics Track Overview". In: (2007). DOI: `https://trec.nist.gov/pubs/trec16/papers/GEO.OVERVIEW16.pdf`.

[71] Xiaoshi Yin, Xiangji Huang, and Zhoujun Li. "Towards a better ranking for biomedical information retrieval using context". In: *2009 IEEE International Conference on Bioinformatics and Biomedicine*. IEEE. 2009, pp. 344–349.

[72] Claire Fautsch and Jacques Savoy. "IR-Specific Searches at TREC 2007: Genomics & Blog Experiments." In: *TREC*. Citeseer. 2007.

[73] Arman Cohan et al. "A discourse-aware attention model for abstractive summarization of long documents". In: *arXiv preprint arXiv:1804.05685* (2018).

[74] Ani Nenkova and Lucy Vanderwende. "The impact of frequency on summarization". In: *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005* 101 (2005).

[75] Günes Erkan and Dragomir R Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization". In: *Journal of artificial intelligence research* 22 (2004), pp. 457–479.

[76] Josef Steinberger and Karel Jezek. "Using latent semantic analysis in text summarization and summary evaluation". In: *Proc. ISIM* 4 (2004), pp. 93–100.

[77] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *Journal of machine Learning research* 3.Jan (2003), pp. 993–1022.

NTNU

Kunnskap for en bedre verden