Master's thesis

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

Martin Stiles

# Entity-Related Multi-Document News Summarization

Master's thesis in Computer Science
Supervisor: Björn Gambäck
June 2022

NTNU
Norwegian University of
Science and Technology

Martin Stiles

# Entity-Related Multi-Document News Summarization

Master's thesis in Computer Science
Supervisor: Björn Gambäck
June 2022

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

NTNU
Norwegian University of
Science and Technology

## Abstract

In this era of information overload, the need for systems that compress data while preserving primary information is ever-growing. Automatic summarization systems aim to do just that – reduce the size of textual documents while preserving the key points. The field of automatic summarization has seen significant growth in recent years, especially with the introduction of transformers. However, one potentially valuable summarization task that seemingly has not received much attention from prior research is *entity-related summarization*. Entity-related summarization encapsulates the task of generating summaries concerning one or more entities based on textual documents. This master's thesis aims to present a novel approach to entity-related summarization in a multi-document environment.

A naive approach to making summaries related to an entity is proposed: given a collection of text documents and an entity, remove all sentences that do not mention the entity and summarize the remaining text with standard summarization methods. Two summarization systems adopting the approach are designed and implemented to evaluate it. The first system is an extractive system focusing on minimizing redundancy, while the second is an abstractive system that utilizes state-of-the-art transformers in a recursive approach.

In order to evaluate the systems, a survey design to collect quantitative data concerning entity-related systems is presented. The design considers the infeasibility of making respondents read the source documents of the summaries. Thus, questions focus on different aspects of the summary texts, like language quality, redundancy, and entity relevance. Further, a survey adopting the aforementioned design is conducted, and substantial results are presented in this thesis.

The abstractive system receives excellent scores from human evaluators, suggesting that it generates summaries of borderline human quality. On the other hand, the extractive model receives decent scores and achieves the goal of minimizing redundancy. However, it seems to struggle with producing coherent summaries. On average, evaluators prefer the abstractive system in 79.3% of cases, showing that the abstractive design is vastly superior to the extractive. All things considered, the results suggest that the proposed approach for entity-related summarization works to a satisfactory degree.

## Sammendrag

I denne tiden med informasjonsoverflod er behovet for systemer som komprimerer data uten å miste den viktigste informasjonen stadig økende. Systemer for automatisk oppsummering sikter på å gjøre nettopp det – redusere størrelsen på tekstdokumenter mens nøkkelpoengene bevares. Automatisk oppsummering av tekst er et forskningsfelt som har hatt betydelig vekst de siste årene, spesielt etter introduksjonen av forhåndstrente, oppmerksomhetsbaserte språkmodeller. En potensielt verdifull oppsummeringsoppgave som tilsynelatende ikke har fått mye oppmerksomhet fra tidligere forskning er *entitetsrelatert oppsummering*. Entitetsrelatert oppsummering omfatter generering av sammendrag som er knyttet til en eller flere entiteter basert på tekstdokumenter. Denne masteroppgaven har som mål å presentere en ny tilnærming til entitetsrelatert oppsummering basert på flere tekstdokumenter.

En relativt enkelt tilnærming til å lage sammendrag relatert til en entitet er foreslått: gitt en samling av tekstdokumenter og en entitet, fjern setninger som ikke nevner entiteten og oppsummer den gjenværende teksten med vanlige oppsummeringsmetoder. To oppsummeringssystemer som tar i bruk tilnærmingen er designet og implementert for å evaluere denne. Det første systemet er ekstraksjonsbasert og fokuserer på å minimere repetisjon. Det andre systemet er abstraksjonsbasert og bruker toppmoderne forhåndstrente språkmodeller med en rekursjonsbasert tilnærming.

For å evaluere systemene er et undersøkelsesdesign for innsamling av kvantitative data vedrørende entitetsrelaterte systemer presentert. Designet tar hensyn til utfordringene med å få respondenter til å lese kildedokumentene til oppsummeringene. Følgelig fokuserer spørsmålene på andre aspekter ved oppsummeringstekstene, som språkkvalitet, redundans og entitet-relevans. Videre er det utført en spørreundersøkelse som tar i bruk det nevnte designet, og betydelige resultater presenteres i denne masteroppgaven.

Det abstraksjonsbaserte systemet mottar meget gode poengsummer fra undersøkelsesdeltakere, noe som antyder at systemet genererer oppsummeringer på grensen til menneskelig kvalitet. Det ekstraksjonsbaserte systemet mottar akseptable poengsummer og oppnår målet om å minimere repetisjon. Systemet ser derimot ut til å ha utfordringer med å produsere sammenhengende oppsummeringer. I gjennomsnitt foretrekker undersøkelsesdeltakere det abstraherende systemet i 79,3% av tilfellene, noe som viser at det abstraksjonsbaserte designet er overlegen det ekstraksjonsbaserte. Med alt tatt i betraktning virker det som at den foreslåtte tilnærmingen til entitetsrelatert oppsummering fungerer til en tilfredsstillende grad.

# Preface

This master's thesis is a conclusion to a five-year master's program in Computer Science at the Norwegian University of Science and Technology (NTNU), Trondheim, and is described by the course TDT4900[1]. The work was conducted in the spring of 2022 at the Department of Computer Science, with Björn Gambäck as a supervisor. A specialization project was conducted in the autumn of 2021 as preparation, and some ideas are incorporated into this thesis. The thesis is written in collaboration with Strise[2] – a Norwegian text analytics company that provided me with annotated real-world data.

First and foremost, I want to give a special thanks to my supervisor Björn Gambäck for always sticking up for me, even in times when my motivation was *somewhat lacking*. Through frequent meetings, Björn has provided me with brilliant guidance and crucial feedback, and he has continuously answered every question I have thrown at him. Björn has also allowed me the freedom to experiment with methods and approaches that I found intriguing, which I am very grateful for. Secondly, I want to thank the people at Strise for the opportunity to work with them on a topic that genuinely interests me. I wish to thank Patrick Skjennum, CTO of Strise, who helped me identify a suitable research problem and get started with the project. Further, I wish to hand out a special thanks to Sigurd Berglann at Strise, who has played an important role in this thesis. Sigurd and I had weekly meetings with updates and discussions, and he has been of tremendous help with regard to creating a relevant dataset for the project. Additionally, Sigurd has given me valuable feedback for the thesis. Lastly, I want to thank my friends and family for encouraging me during the writing process and everyone who was kind enough to respond to the survey. A special thank you to my friend Stefan for helping me test the survey and for numerous exciting discussions about relevant topics.

Martin Stiles
Trondheim, 24th June 2022

[1] https://www.ntnu.edu/studies/courses/TDT4900#tab=omEmnet
[2] https://www.strise.ai/

# Contents

# List of Figures

# List of Tables

# 1. Introduction

This chapter intends to introduce the reader to the content of this master's thesis. The following sections present the motivation for the chosen field of study (Section 1.1), the goals and research questions that form the basis for the work (Section 1.2), the plan to achieve the goal and answer the questions (Section 1.3), the contributions (Section 1.4), and the overall structure of the master's thesis.

## 1.1. Background and Motivation

The amount of textual data on the internet (e.g., news articles and blog posts) has been rapidly increasing in the past decade, and no trends indicate that this growth will slow down in the near future. Consider, for instance, that over five million new blog posts were published every day on WordPress in the first quarter of 2021 (71% of which were in English) (Chernev, 2022), or consider that there are 3,623 active newspaper publishing businesses in the US alone as of January 2022.[1] For humans, staying up-to-date with such large amounts of new information is impossible; there are too many sources and articles to choose from. Thus, the need for sophisticated summarization systems that can compress large bodies of text into a short, human-readable format is at an all-time high. Automatic summarization has even been considered one of the most important yet least solved tasks within the field of Natural Language Processing (Gidiotis and Tsoumakas, 2020).

Summarization is the act of reducing the size of a document/collection of documents while preserving the critical information of the source. Recent years have brought much progress to the field of automatic summarization, especially with the introduction of transformer models by Vaswani et al. (2017). One aspect of summarization that, seemingly, has not received much attention from prior research is *entity-related summarization based on textual data*. "Entity-related" implies that a summary concerns one or more entities, and "based on textual data" implies that source documents for the summary are purely text-based. An example task that this thesis aims to solve is: have 20 articles about the storming of the capitol on January 6th, 2021, and then generate a summary related to Donald Trump. A *normal* summary would focus on what happened at the capitol. However, an *entity-related* summary (with Trump as the entity) would focus on Trump's involvement more than anything else.

The idea behind this thesis was devised in collaboration with Strise.[2] Strise works with

[1] https://www.ibisworld.com/industry-statistics/number-of-businesses/newspaper-publishing-united-states/

[2] https://www.strise.ai/

closely related topics, and their interest in entity-related multi-document summarization shows that the task may have value from a business standpoint. Thus, the motivation is not only to contribute to the field of automatic summarization, but also to provide a novel approach to a task that could be of value to others.

## 1.2. Goals and Research Questions

The aim is to contribute to the field of automatic summarization by researching the task of entity-related multi-document summarization. "Entity-related" implies that a summary concerns a specific entity. "Multi-document" implies that a summary is based on more than one text document.

**Goal** *Provide a novel approach to automatic entity-related summarization based on a collection of text documents.*

As the literature review identified no previous research describing systems for entity-related summarization directly from textual data, the goal is to provide and test a naïve approach. Given an input collection of text documents and an input entity, the naïve approach is to remove sentences that do not mention the entity and summarize the remaining texts as typical summarization methods do. The approach is mainly intended for multi-document summarization, but the general idea should also apply to single-document summarization. The central idea is to create two summarization systems that implement this naïve approach and then create a basis for evaluation by conducting a survey.

**Research question 1 (RQ1)** *Does removing sentences that do not include a specific entity and using a state-of-the-art summarization model to summarize the remaining text produce a good entity-related summary?*

As mentioned, the approach to making entity-related multi-document summaries revolves around removing sentences that do not mention the specific entity. As a consequence of removing sentences, the remaining text documents only consist of sentences that are relevant to the entity. This begs the question: does summarizing the remaining sentences with a state-of-the-art summarization model result in a summary that captures the main information about the entity? RQ1 is the central question of the thesis, and the primary basis to answer this question is the survey conducted in Section 6.3.

**Research question 2 (RQ2)** *Can the summarization systems be considered state-of-the-art for automatic summarization tasks?*

RQ1 specifies "using a state-of-the-art summarization model". Although the implemented summarization systems are inspired by state-of-the-art systems, they are modified to fit the desired use case. Thus, a thorough evaluation of the systems is necessary to conclude whether the systems can be regarded as state-of-the-art.

**Research question 3 (RQ3)** *Are the summarization systems suitable for a real-world scenario?*

Two summarization systems are presented in this thesis: one extractive (Section 5.3) and one abstractive (Section 5.4). It is desirable to compare the systems and discuss whether they are suitable for a real-world scenario. A "real-world scenario" refers to when the system is not just run on a predetermined dataset, but used to create some sort of value, for instance on a website that generates summaries of multiple news articles on the fly. In such cases there are more aspects to account for than just summary quality, like time consumption and consistency.

**Research question 4 (RQ4)** *Are the summaries produced by the summarization systems satisfactory according to human evaluators?*

Although RQ4 might seem similar to RQ1, RQ1 aims to find out whether the idea of removing irrelevant sentences works. In contrast, RQ4 aims to determine whether the summarization systems produce acceptable summaries regarding different aspects of the summary text. Evaluating summarization systems is arduous, and human evaluation is often necessary. Thus, the survey presented in Section 6.3 will create the basis for answering RQ4.

## 1.3. Research Method

The methodology adapted in the first half of this thesis is primarily theoretical. At first, a literature review is conducted on multi-document summarization, as no related work was identified for entity-related summarization. Note that a structured literature review of a similar field, long text summarization, was conducted in a specialization project to prepare for the master's thesis. Therefore, the related work identified in the specialization project will be incorporated into the thesis literature review. Further, the summarization systems are implemented, and the theoretical part of the work concerns designing the systems, as some assumptions about the approaches must be made. A survey is the main basis for evaluating the systems, meaning an exploratory approach for the system designs is practically infeasible.

In contrast to the first, the second half of the thesis is primarily analytical. Experiments are conducted to gather empirical data regarding the performance of the summarization systems. Furthermore, the results are utilized for discussion and evaluation. Thus, there should be a solid basis for answering the research questions.

## 1.4. Contributions

The main contributions, which are described in more detail in Section 8.2, can be summarized as follows:

- An approach to generate entity-related summaries from a collection of texts by naïvely removing sentences that do not mention the entity and then summarizing the remaining sentences in a typical manner.
- The design and implementation of two entity-related multi-document summarization systems: one extractive with a focus on minimizing redundancy and one abstractive that uses state-of-the-art transformer models.
- A survey design that provides a framework for evaluating entity-related summarization systems, along with results from a conducted survey.
- A thorough evaluation of the systems based on results from the survey, which shows that the abstractive system demonstrates close to human-level performance.

## 1.5. Thesis Structure

This section lists all chapters along with a short description of the intention behind each chapter.

1. **Introduction**
   Provides introductory information about the field of study and presents the thesis's goal and main contributions.
2. **Background Theory**
   Introduces topics and technical concepts relevant for automatic summarization and thus for this thesis.
3. **Related Work**
   Explains that no related work within the field of entity-related summarization from text has been identified. Consequently, prior research within the field of multi-document summarization is reviewed and presented instead.
4. **Data**
   Gives a short presentation of relevant datasets, including the working dataset, which has been designed and created solely for this project.
5. **Architecture**
   Presents the design and implementation of two summarization systems, including the data preparation step that enables the systems to generate entity-related summaries.
6. **Experiments and Results**
   Presents how a dataset was used to evaluate the systems automatically, how a survey was designed and conducted, and the relevant results.

7. **Evaluation and Discussion**
   Provides an evaluation of the summarization systems, discusses limitations of the experiments and the systems, and revisits the research questions and the goal in light of conducted work.

8. **Conclusion and Future Work**
   Intends to conclude the thesis by summarizing the work, presenting the most important contributions, and suggesting ideas for future work within the field of entity-related summarization.

# 2. Background Theory

This chapter aims to describe concepts that are relevant throughout this thesis. Concepts primarily regard summarization and technological topics related to Natural Language Processing (NLP). Many of the following concepts are intricate and require extensive reading to fully understand, but this chapter only intends to familiarize the reader with the concepts. Thus, this chapter does not explain concepts in greater detail than what is deemed necessary to continue reading the thesis. For instance, the underlying mathematics of attention mechanisms will not be covered, but the general idea will be explained. Finally, it should be mentioned that some descriptions are adapted from the specialization project leading up to the thesis.

## 2.1. Summarization Fundamentals

Text summarization is the act of making a short textual representation of one or more text documents while preserving the crucial information of the source. Automatic text summarization thus refers to a computer program that takes textual document(s) as input and produces a summary. The following paragraphs present the main types of automatic summarization.

#### Extractive vs Abstractive

Extractive summarization, as the name suggests, concerns identifying the most important sentences in the source document(s) and then concatenating them to form a summary. Producing extractive summaries is, in essence, a selection problem. Consequently, extractive methods can be very efficient and do not have to be sophisticated. Furthermore, sentences in a summary are extracted from human-written text and thus guaranteed to be of human quality. However, a common problem is poor sentence coherence (Wu and Hu, 2018), as there is no guarantee extracted sentences fit together.

In contrast to extractive, abstractive methods can generate sentences that do not exist in the source document(s). Consequently, abstractive methods require a deep semantic understanding of the source and a way of generating text. Abstractive methods resemble the way humans summarize, and by nature, the potential for such methods is much higher than for extractive. However, abstractive methods are usually complex and are prone to grammatical errors or fake fact generation (Cao et al., 2018)

### Single-Document vs Multi-Document

Although rather self-explanatory, single-document summarization (SDS) is when there is only one source document, while multi-document summarization (MDS) is when there is more than one. SDS approaches are more researched and generally more straightforward to implement than MDS approaches. In addition, MDS approaches tend to be more susceptible to including repetitive sentences in summaries, as more redundant information exists across the documents (Calvo et al., 2018).

## 2.2. Natural Language Processing Concepts

This section briefly describes NLP concepts that the reader should be familiar with.

### Data Cleaning

Data cleaning comprises a plethora of methods to make data more suitable for other NLP techniques. The following list presents common steps for data cleaning:

- *Tokenization* is the process of dividing a text into smaller units called tokens. Tokens are usually words or sentences.
- *Stop-word removal* is the process of removing common words that do not add value to the text besides grammatical structure. In English, examples of stop-words are "the", "of", and "at".
- *Punctuation removal* is the process of removing punctuation, e.g., question marks, commas, and parentheses.
- *Stemming* is the process of reducing a word to its root (or stem). This is usually done to make different variants of a word equal. For instance, after stemming, both "study" and "studying" becomes "studi".

### *n*-grams

An $n$-gram is a tuple consisting of $n$ words. A 1-gram (unigram) is just a single word, a 2-gram (bigram) contains two words, and so on. Different n-grams can be used to represent text. For instance, consider the sentence "dogs are cool". A unigram representation would be ["dogs", "are", "cool"], while a bigram representation would be ["dogs are", "are cool"].

### Longest Common Subsequence

The longest common subsequence in NLP refers to the longest subsequence of words common to all given text sequences. LCS is perhaps best explained by an example, so consider the following sentences: "**I** love **my family**" and "**I** visited **my family** yesterday". The longest subsequence of words appearing in both sentences is ["I", "my", "family"], which is thus the LCS.

### Semantics Meaning

Semantic in NLP refers to what a text or a sentence actually means. Thus, two sentences are semantically similar if they convey similar information, despite potential differences in word usage. For example, consider the following sentences: "my neighbor John has a dark red car" and "the person who lives next to me owns a wine-colored automobile." The two sentences convey the same information but have no words in common other than "a." Classical NLP measures for sentence similarity (e.g., n-gram overlap) are infeasible in such cases.

### Vector Representations

It is desired to represent sentences as vectors to enable the use of mathematical operations. Traditional methods for converting sentences into vectors, like the bag-of-words (BOW) model, are straightforward. The BOW model constructs a vocabulary based on every word in the given sentences, which represents the vector shape. Each sentence is then represented by a vector where each number is the word count of the associated word in the vocabulary. On the other hand, modern approaches use sophisticated models like neural networks trained to encode sentences into a fixed-size vector. Neural-network-based models are good at capturing the semantic meaning of sentences. An example of such a model is Sentence-BERT (Reimers and Gurevych, 2019), which is used later in this thesis.

### Cosine similarity

Cosine similarity is a standard measure for vector similarity. Consequently, it can be applied to sentence vectors to measure sentence similarity. Given two n-dimensional vectors, A and B, cosine similarity is calculated by the following equation:

$$cosine\_similarity(A, B) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

### Sentence Relevance Scoring

A sentence relevance score can be used to find the most important sentences in a collection of sentences. The sentence with the highest score can thus be deemed the most representative in the collection. This thesis implements a sentence relevance algorithm inspired by TextRank (Mihalcea and Tarau, 2004). TextRank, in essence, calculates the similarity between all sentences and thus ranks sentences according to the similarity measurements.

### Named Entity Recognition and Coreference Resolution

Named Entity Recognition (NER) is the process of identifying and classifying entities in text. Entities include persons, places, organizations, and more. Classifying entities are

not always straightforward. For instance, consider the mention "Orange": does it refer to the color, or the fruit of an orange tree? Thus, NER systems require sophisticated designs, as they often have to account for the context of mentions. Coreference Resolution (CR) is similar to NER in that it classifies entities, but the goal of CR is to identify mentions that refer to the same entity. A major difference from NER system is that CR systems intends to classify non-named mentions, for instance, "she", "it", or "they". Consequently, NER and CR are often used in combination.

#### Language Models and Language Generation

Language models are probability distributions designed to calculate the likelihood for following words based on a text sequence. For instance, given the sequence "once upon a", the word "time" should likely appear next. A primary application for language models is language generation. The model will effectively generate new text by iteratively predicting the next word based on a sequence and adding it to the sequence. Popular methods for choosing the next word for a sequence are top-k and top-n sampling (nucleus sampling) (Holtzman et al., 2019).

## 2.3. Machine Learning

Machine learning is a branch of artificial intelligence devoted to developing systems that can identify patterns in data with minimal human involvement. Thus, machine learning systems intend to *learn* from data and then utilize the learning process for future decisions. This section presents two primary types of machine learning, namely unsupervised learning and supervised learning.

### 2.3.1. Unsupervised Learning

Unsupervised learning comprises algorithms that, given a set of mathematical instructions, can recognize patterns in data without further human intervention. Unsupervised methods do not rely on extensive training with labeled data, making them convenient for tasks with no labeled datasets available. The perhaps most popular branch of unsupervised learning concerns clustering problems.

#### Clustering

Clustering is the process of grouping similar data points to discover inherent groupings in the data. During clustering, every data point is assigned a class. The underlying idea is that data points that end up with the same class belong to the same group. To group data points, a measure of similarity must be applied.

#### K-Means Clustering

One of the most common clustering algorithms is K-means. K-means is an iterative approach using centroids to represent the clusters. A centroid is a data point in the same

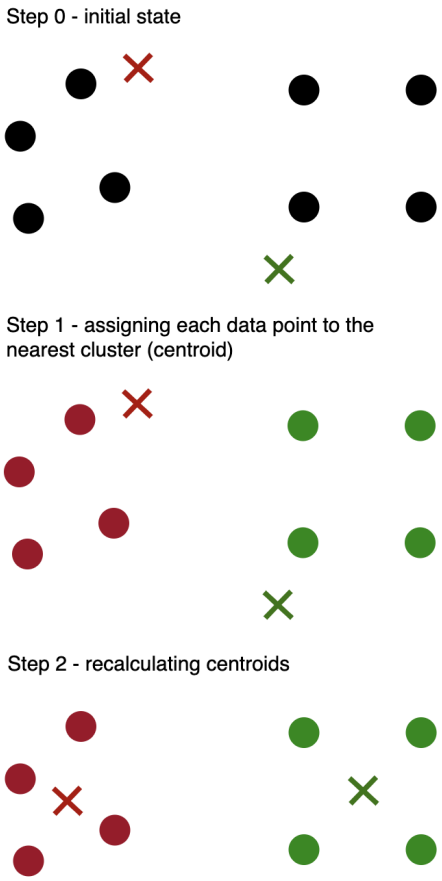



Figure 2.1.: Illustration of K-means with K = 2. Step 0 shows how eight data points are distributed, and two random centroids (red and green) are initiated. Steps 1 and 2 illustrate the iterative process of assigning data points to clusters and recalculating centroids.

vector space as the original data. Centroids are usually initiated with random values. Furthermore, to run K-means, it is required to specify K: the number of clusters/centroids. Thus, the iterative process of K-means is as follows:

1. Assign every data point to the nearest cluster. The nearest cluster is found by measuring the similarity between the data point and every centroid.

2. For each cluster, calculate the average of the data points assigned to it. This average then becomes the new centroid to represent the cluster.

Figure 2.1 illustrates this iterative process. K-means is usually stopped when newly calculated centroids are very similar to the previous centroids, or if the algorithm reaches the maximum number of iterations.

### Silhouette Coefficient

Finding the optimal value of K for K-means is complicated and often requires domain-specific knowledge. However, the optimal number of clusters often changes from case to case, demonstrating the need for an automatic method to approximate K. A silhouette coefficient is a commonly used measure of how good a given clustering is.

The silhouette coefficient for a particular value of K is the average silhouette score of each data point. A data point's silhouette score is calculated by using its *cohesion* and its *separation*. Cohesion, $a$, is the mean similarity between the data point and every other data point in its cluster. Separation, $b$, is the mean similarity between the data point and every data point in the next nearest cluster. Thus, the silhouette score of a data point is calculated by:

$$S = \frac{b - a}{max(a, b)}$$

To approximate the optimal value of K, the silhouette coefficient of the clustering from a range of K values is calculated. A higher silhouette coefficient implies better clustering; thus, the value of K resulting in the highest silhouette coefficient is chosen.

## 2.3.2. Supervised Learning

In contrast to unsupervised learning, supervised learning relies on labeled data to train models for a specific task. The idea behind supervised learning is to give computer programs the necessary tools to figure out how to perform a task by themselves. The upside of this approach is that it works well for tasks that are arduous to solve with traditional programming, e.g., image classification. However, one of the major downsides of supervised learning is that it requires human-labeled datasets, which can be hard to obtain. Further, it is common for supervised models to become a "black box", meaning that it is hard or even impossible to understand why the model produces certain results, raising ethical questions. The following paragraphs present neural-network-based approaches to supervised learning leading up to transformer models, which are highly relevant for this thesis.

### Artificial Neural Networks

The design of artificial neural networks (ANN) is inspired by the human brain, especially regarding how biological neurons function together. The perceptron is the essential building block to achieve this, illustrated in Figure 2.2. A perceptron has multiple inputs, and each input has an associated weight. The output from the perceptron is calculated by an activation function, which is usually the weighted sum of the inputs (hence the summation sign in Figure 2.2).

Perceptrons combined in a network with a layer structure compose a feed-forward neural network (FNN), as illustrated in Figure 2.3. FNNs consist of one input layer, one output layer, and at least one hidden layer. Neural networks are called deep neural

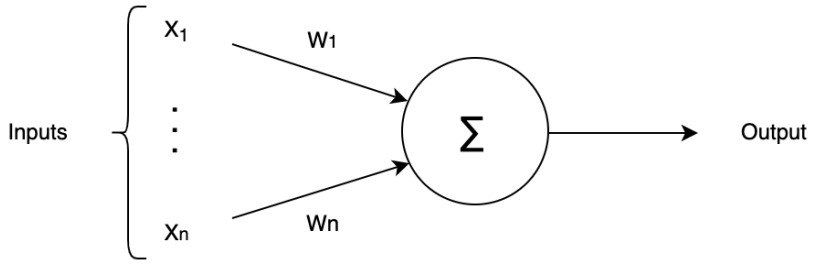



Figure 2.2.: Illustration of a perceptron.

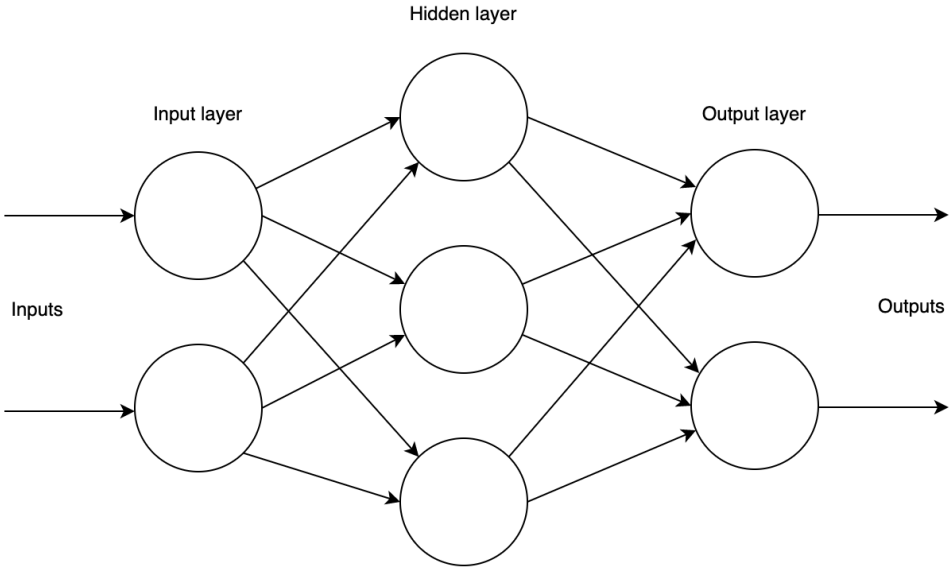



Figure 2.3.: Illustration of a feed-forward neural network.

networks if they contain more than one hidden layer. FNNs are the simplest form of ANNs as information only flows in one direction; forward (from input to output).

A FNN intends to approximate a function. This is achieved by having perceptrons with adjustable weights in the network. When given labeled data, the objective of the FNN becomes a minimization problem: adjust the weights in the network to minimize the deviation between the produced output and the desired output. In this way, the FNN simulates learning.

### Recurrent Neural Networks

A recurrent neural network (RNN) is a type of ANN designed for sequential data. "Sequential" implies that the ordering of the data is crucial and data points are not independent. Thus, an example of sequential data is textual data, as words depend on the sentence in which they appear. The design of RNNs is often seen as the opposite of FNNs, as RNNs allow for backward information flow by adding loops. Consequently, RNNs can utilize past information, effectively achieving something akin to memory. However, it is insufficient to store numerous previous data points due to memory limitations. Thus, RNNs struggle with long inputs. Further, RNNs suffer from the vanishing gradient problem, meaning that the gradient storing past information will get smaller over time, and consequently, the information disappears. It is also worth noting that RNNs have a sequential dependency on the input data, which means they cannot be parallelized.

### Sequence-to-Sequence Models

Sequence-to-sequence (seq2seq) models, introduced by Sutskever et al. (2014), adapt an encoder-decoder architecture. The encoder-decoder architecture consists of two main components: an encoder that sequentially encodes the input sequence into a context vector and a decoder that sequentially generates an output sequence based on the context vector. This process is illustrated in Figure 2.4, where an input sequence of length $n$ is transformed into a context vector and an output vector of size $m$. The encoder and the decoder usually comprise multiple RNN-based models.

Similar to RNNs, seq2seq models struggle when the input length (n) is large. This problem arises due to having a fixed-length context vector, which means that the decoder has limited access to information from the input. Thus, when the input is large, the information exceeds what can be stored in the context vector, and the decoder misses out on potentially important details. In Figure 2.4, this happens in hidden state $h_n$, as the final hidden state cannot contain all the previous information. This is a fatal flaw for applications like automatic summarization, where inputs are expected to comprise numerous sentences.

### Attention Mechanisms

Attention mechanisms were inspired by attention by the human visual processing system and first introduced by Larochelle and Hinton (2010) for computer vision. However, it was

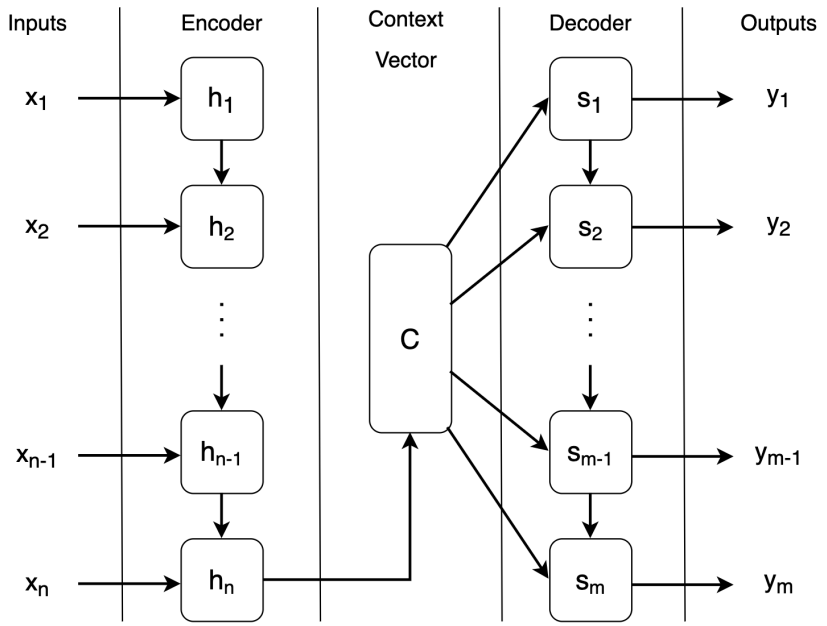



Figure 2.4.: Illustration of a sequence-to-sequence model.

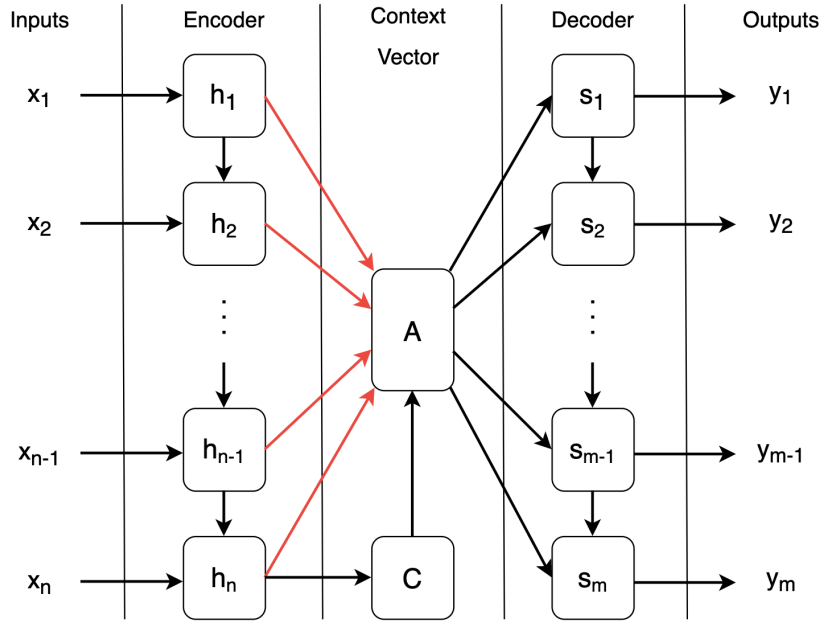



Figure 2.5.: Illustration of a sequence-to-sequence model incorporating attention mechanisms. Red arrows represent the attention mechanisms.

later introduced to the field of NLP by Bahdanau et al. (2014) for a machine translation task. The key idea behind attention mechanisms is to improve the performance of seq2seq models when given long sequential inputs, an issue discussed in the previous paragraph. Attention mechanisms solve this problem by passing all hidden states to the encoder, as illustrated in Figure 2.5. The decoder can thus score the hidden states according to their relevance, effectively giving more attention to relevant parts of the input sequence.

#### Transformers

Transformer models, introduced by Vaswani et al. (2017), utilize an encoder-decoder architecture like other seq2seq models. However, as the paper's name suggests ("Attention is All you Need"), transformers incorporate nothing more than attention mechanisms (presented in the previous paragraph). Thus, transformers differ from traditional seq2seq models by removing the sequential dependency on the input, making transformers fully parallelizable. Consequently, transformers can be trained much more efficiently, and it is feasible to train them on massive datasets.

Numerous pre-trained language models utilizing the transformer architecture have been released after the introduction of transformers. Such pre-trained transformers are usually huge models trained on massive domain-general text data. Consequently, they show an unrivaled ability to understand natural language. Furthermore, they can be fine-tuned on domain-specific data to adapt to a particular task. Note that these transformers are semi-supervised since they are first pre-trained in an unsupervised manner on a large unlabeled dataset and then fine-tuned in a supervised manner on a domain-specific labeled dataset.

Since pre-trained transformers are state-of-the-art language models, handle relatively long inputs well, and can be fine-tuned for specific tasks, they are well-suited for automatic summarization. PEGASUS (Zhang et al., 2019) is a pre-trained transformer used for summarization in this thesis. PEGASUS's pre-training objective is intentionally similar to summarization: important sentences from the input are removed and then generated as one output sequence based on the remaining input sentences. Thus, the pre-training objective simulates abstractive summarization by making the transformer generate extractive summaries word by word. As a result, PEGASUS achieves state-of-the-art performance on numerous downstream summarization tasks.

## 2.4. Evaluation Metrics

This section introduces evaluation metrics relevant to the results presented in this thesis.

### 2.4.1. Precision, Recall and F1-score

Precision and recall are commonly used metrics for evaluating machine learning systems. Given a set of predictions, precision and recall are calculated using the following equations:

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

F1-score is the harmonic mean of precision and recall, which implies that it is preferable to have two medium scores instead of one good and one bad. For instance, $precision = recall = 0.5$ results in a better F1-score than $precision = 0.1$ and $recall = 0.9$. F1-score is calculated by the following equation:

$$\text{F1-score} = \frac{2 * \text{P} * \text{R}}{\text{P} + \text{R}}$$

### 2.4.2. ROUGE

Recall-Oriented Understudy of Gisting Evaluation (ROUGE), introduced in Lin (2004), adapts the idea of precision, recall, and F1-scores to work with text. ROUGE describes a set of evaluation metrics primarily used for evaluating summarization and translation systems. Unless stated otherwise, one usually refers to the ROUGE F1-score when discussing ROUGE scores. The most commonly used versions of ROUGE are ROUGE-n and ROUGE-L, which are explained further in the following paragraphs.

#### ROUGE-n

ROUGE-n refers to the n-gram overlap between a system-generated text and the associated reference text. ROUGE-n precision and recall scores are calculated using the following equations:

$$\text{ROUGE-n (precision)} = \frac{\text{num overlapping n-grams}}{\text{num n-grams in reference text}}$$

$$\text{ROUGE-n (recall)} = \frac{\text{num overlapping n-grams}}{\text{num n-grams in generated text}}$$

Further, the ROUGE-n F1-score is calculated by the harmonic mean of ROUGE-n precision and ROUGE-n recall, similar to the F1-score in 2.4.1. It should be mentioned that the most common values for $n$ are 1 and 2. ROUGE-1 measures word overlap, and ROUGE-2 measures bigram overlap.

#### ROUGE-L

ROUGE-L, or ROUGE-LCS, utilizes the length of the longest common subsequence (LCS) between a system-generated text and the associated reference text. LCS is explained further in Section 2.2. The main advantage of ROUGE-L over ROUGE-n is that it does not depend on consecutive n-gram matches and may thus capture sentence structure

more accurately. ROUGE-L precision and recall scores are calculated using the following equations:

$$\text{ROUGE-n (precision)} = \frac{\text{LCS(generated, reference)}}{\text{num words in reference text}}$$
$$\text{ROUGE-n (recall)} = \frac{\text{LCS(generated, reference)}}{\text{num words in generated text}}$$

Once again, the ROUGE-L F1-score is calculated by the harmonic mean of ROUGE-L precision and ROUGE-L recall.

## 2.5. Python for Natural Language Processing

This section briefly presents Python libraries and resources for NLP mentioned in this thesis.

**scikit-learn** scikit-learn[1] is an open-source machine learning library. It provides, for instance, an implementation of the K-means clustering algorithm, including a built-in silhouette score method.

**NLTK** Natural Language Toolkit[2] (NLTK) is a library containing set of tools for NLP. Most relevant for this thesis are simple data cleaning methods, e.g., a sentence tokenizer.

**rouge-score** rouge-score[3] is a library providing a native implementation of ROUGE. Thus, it allows easy calculation of ROUGE-1, ROUGE-2, and ROUGE-L scores.

**Hugging Face** Hugging Face[4] is an AI community built around open-sourcing machine learning models and other tools primarily used for NLP. The Hugging Face API can be accessed through python libraries, and relevant for this thesis are the transformer library and the dataset library. The transformer library grants access to state-of-the-art transformer models (e.g., PEGASUS), and the dataset library grants access to popular NLP datasets.

[1] https://scikit-learn.org/
[2] https://www.nltk.org/
[3] https://pypi.org/project/rouge-score/
[4] https://huggingface.co/

# 3. Related Work

This chapter aims to delve into prior research related to the task of this thesis. Section 3.1 presents the problem of identifying related work and why finding an alternative approach was necessary. Section 3.2 presents the alternative approach taken and aims to familiarize the reader with common approaches to multi-document summarization, as well as the de facto methods for evaluating summarization systems.

## 3.1. Initial Plan

Initially, the plan was to perform a structured literature review (SLR) in correspondence with the methodology presented in Kofod-Petersen (2015), similar to what was done in the specialization project prior to this thesis. The goal of the SLR was to identify related work regarding entity-related summarization based on text. However it was discovered that "entity summarization" refers to summarization of linked data (Liu et al., 2021, 2020; Wei et al., 2019). Thus, the literature search was cluttered by entity summarization approaches and the search to identify any prior work for entity-related summarization based on text.

The field deemed most similar to the task in this project was query-based summarization (QBS), and a possibility was to perform a SLR regarding QBS. However, after inspecting QBS resources, e.g., the AQuaMuSe dataset (Kulkarni et al., 2020), it was discovered that queries mainly consist of questions. Thus, QBS is more akin to a question-answering system and does not really align with the intent of this thesis. Consequently, finding an alternative approach to the SLR was considered necessary.

## 3.2. Alternative Approach

Since no prior research for entity-related summarization based on textual data was identified, a naïve approach to the task was proposed: remove sentences that do not mention the relevant entity and perform summarization as normally on the remaining sentences. With this approach, everything regarding the entity-related part would happen during data preparation. Data preparation and summarization are two independent parts of the system, and consequently, a literature review could be conducted for the individual parts. However, the process of finding entity-mentions is trivialized by the fact that Strise's[1] contribution to this thesis is a sophisticated Named Entity Recognition system. A dataset with source documents and entity mentions was created for this thesis

[1]The company that the thesis project was done in collaboration with – https://www.strise.ai/

and is presented in detail in Section 4.1. Thus, removing sentences that do not mention a specific entity is straightforward and does not require a literature review.

Identifying related work within the field of multi-document summarization (MDS) is desirable, as a part of the thesis goal is to create two summarization systems. Thus, identifying MDS methods is crucial regarding the system designs. This section presents how related work was identified and relevant findings.

### 3.2.1. Methodology

Automatic summarization is a large research field, and the number of approaches to solve the task is seemingly endless. A search for "multi-document summarization" on Google Scholar[2] returns 135 000 research papers. Thus, the scope must be narrowed down, and it is decided to pursue two types of MDS: clustering-based approaches and approaches that enable the use of SDS systems for MDS. Consequently, the two approaches for MDS are explored in the following sections. Additionally, evaluation methods for automatic summarization are explored because evaluating the summarization systems is a key part of the project.

An approach must be taken to identify suitable papers. In the chosen methodology, the first step is to identify survey papers for automatic summarization. Survey papers organize and present relevant work within the field, and three main survey papers concerning automatic summarization were identified for the literature review (El-Kassas et al., 2021; Klymenko et al., 2020; Nazari and Mahdavi, 2019). There is a lot of useful information in the survey papers, but just as important is the fact that they contain a lot of relevant citations. Thus, the next step is to employ a snowballing approach to inspect citations of already selected papers. The survey papers are used as a starting point for snowballing.

Lastly, it should be mentioned that a structured literature review on the field of long text summarization (LTS) was conducted in a specialization project serving as preparation for this thesis. LTS and MDS are similar in that they face a similar problem: the input size is too large for typical encoder-decoder approaches, which currently dominate the field of single-document summarization (Lebanoff et al., 2018). Thus, some findings from the specialization project are included in this literature review.

### 3.2.2. Clustering-Based Multi-Document Summarization

Clustering-based summarization is based on the idea of grouping sentences by using a similarity measure, separating sentences into clusters regarding similar topics, and identifying dominant topics in the process (Nazari and Mahdavi, 2019). Such approaches are well-suited for MDS because they can avoid redundant information in summaries (El-Kassas et al., 2021). However, popular clustering algorithms require a predetermined value for the number of clusters, and estimating the optimal number is not a straightforward task (El-Kassas et al., 2021). For instance, deciding the optimal K in K-means is arduous when dealing with semantic data (M. Salih and Jacksi, 2020).

[2] https://scholar.google.com/

### Clustering-Based Approaches

Zhao et al. (2020) present a clustering-based abstractive system achieving state-of-the-art results on the multi-news dataset. They divide the process into four steps. The first step is text processing, which mostly consists of separating the sentences in the source documents. Minimal processing is applied to keep the sentences raw for subsequent processing. The second step is to build a sentence graph, where nodes are sentences, and edges represent lexical and semantic relations between sentences. The third step is to cluster the sentences by defining a feature vector for each sentence based on the graph (Zhao et al., 2020) and then applying K-means. Lastly, an abstractive summary is generated for each cluster using sentence compression, and the concatenation of those summaries comprises the final summary.

Nazari and Mahdavi (2019) specifically present a four-step process for clustering-based summarization, consisting of pre-processing, sentence clustering, cluster ordering, and selecting representative sentences from clusters. This process is similar to what is done by Zhao et al. (2020), but clustering is performed directly on sentence representations instead of constructing a graph first.

Bouscarrat et al. (2019) present an efficient extractive approach for summarization by utilizing sentence embeddings to capture the semantic meaning of sentences. The extracted summary is made up of sentences close to a vector representation of the complete source, utilizing sent2vec (Pagliardini et al., 2018) for similarity measurements.

### Clustering Algorithms

M. Salih and Jacksi (2020) conclude that K-means is the most popular algorithm for clustering sentences based on semantic similarity. They note that K-means generates strong clusters as long as the distribution of data points is not too complicated. They also note that hierarchical clustering methods are straightforward to implement (similar to K-means) but often inferior to K-means.

Abdulateef et al. (2020) apply K-means for two different purposes: first for filtering out similar documents, and then, after dividing the remaining documents into sentences, to group similar sentences and extract key sentences to overcome the redundancy problem of MDS systems. They use sent2vec (Pagliardini et al., 2018) to represent sentences as a vector before clustering them with K-means. Others, like Suryadjaja and Mandala (2021), use pre-trained transformers for sentence encoding, stating that models like SentenceBERT (Reimers and Gurevych, 2019) is state-of-the-art for sentence embedding tasks.

## 3.2.3. Adapting Single-Document Systems to Multi-Document Environments

Most previous research within the field of automatic summarization concerns SDS (El-Kassas et al., 2021). SDS methods, especially since the introduction of transformers by Vaswani et al. (2017), have achieved extraordinary results on summarization tasks.

However, as mentioned, one of the problems of MDS is that SDS methods are generally not directly translatable to MDS. This section presents some ideas for using SDS models in a multi-document environment. Note that some research is critical to such approaches – suggesting that using SDS methods may not lead to ideal results (Zhou et al., 2021).

### Transformers for Single-Document Summarization

Large pre-trained transformer-based models dominate the state of the art for SDS (Lebanoff et al., 2018), as can be seen on the benchmark overview for summarization systems on Papers With Code.[3] Examples of such transformers that have seen great results for summarization tasks are BART (Lewis et al., 2019), PEGASUS (Zhang et al., 2019) and GPT-3 (Brown et al., 2020). Transformers are good at considering relatively large text sequences, as described in Section 2.3.2. Pre-trained transformers usually have a maximum input size of 512 or 1024 words. However, multi-document collections can be much longer than 1024 words. Thus, it is undesirable to directly apply a pre-trained transformer to summarize entire collections, as it may only consider a tiny part of the input.

### Recursive Approaches

The idea behind recursive approaches is to break down the input into smaller parts such that SDS models can be applied without truncating the input. For MDS, one part can just be one document, or the documents can be combined and then split into chunks that fit into SDS models. The smaller parts are summarized separately and may be summarized multiple times if necessary.

Wu et al. (2021) propose to do the recursion in a tree-like structure. First, the input is divided into many small parts representing the leaf nodes, and each node is summarized. Then, the recursive step happens by concatenating a subset of the nodes (e.g., three and three nodes) and summarizing the concatenation, creating new nodes higher in the tree. This step is repeated until only one node is left to consider, which becomes the root node. Thus, the root node represents a summary of the entire source text. GPT-3 (Brown et al., 2020) is used at the summarization steps.

Sinha et al. (2018) propose a straightforward recursive approach for an extractive summarization system. Firstly, the input is divided into many small parts, and each part is summarized. However, instead of using a tree-like recursion structure like Wu et al. (2021), all summaries are concatenated and exposed to a length check. The process is repeated if the summary is longer than the maximum allowed length. This means that the concatenation of summaries is again broken into smaller parts, summarized, concatenated and exposed to the length check. A simple feed-forward neural network is used to summarize each part.

Wu et al. (2021) report that summarization errors (e.g., grammatical errors or fake fact generation) tend to propagate through the recursive steps, as noted by human evaluators

[3] https://paperswithcode.com/task/text-summarization#benchmarks

who rated their summarization system at different levels of recursion. This means that if a summarization model produces an error in the early recursive steps, it will likely make the same error in consecutive steps.

#### Transformers With a Large Input Size

The main reason pre-trained transformers normally have a maximum input length of 512 or 1024 words is because of time constraints (Zaheer et al., 2020). Standard pre-trained transformers have a quadratic dependency on the input, making them too slow if the input size increases. However, Zaheer et al. (2020) present BigBird, a model with sparse attention mechanisms that removes the quadratic dependency of full attention mechanisms. Thus, BigBird remains efficient despite increasing the input size to 4096 words. Consequently, it could be used directly for MDS, but it still cannot handle inputs with more than 4096 words without truncation.

### 3.2.4. Evaluating Summarization Systems

Evaluating summarization systems is a very challenging task (El-Kassas et al., 2021), but attempts have been made to automate the task. Every paper concerning a summarization system identified in this literature review uses ROUGE, described in Section 2.4.2, as the baseline evaluation metric. More specifically, every paper uses ROUGE-1, ROUGE-2, and/or ROUGE-L. Thus, there is no doubt about ROUGE being the most common metric for automatically evaluating summarization systems, which is also stated by other papers (Wang et al., 2017; El-Kassas et al., 2021). Owczarzak et al. (2012) show that using ROUGE-2 without removing stop-words aligns the most with human evaluators. However, ROUGE is flawed by design (Stiennon et al., 2020), and human evaluation is often desirable.

#### Limitations of ROUGE

ROUGE intends to measure the similarity between machine-generated and human-written summaries. ROUGE may thus give a good indication of the amount of salient information included in a summary. However, ROUGE represents a misalignment between the optimization problem (achieving a high ROUGE score) and what is actually desirable according to human evaluators (making a good summary) (Stiennon et al., 2020). In short, ROUGE does not necessarily say whether a summary is good or not in terms of what humans deem important. Since ROUGE is based on exact word matches, it will, for instance, give lower scores to generated summaries that use different words (although with similar meaning) than the gold-standard summary (Stiennon et al., 2020). Important aspects regarding summary quality that are not captured by ROUGE include fluency, grammaticality, and coherence.

### Human Evaluation

Knowing that ROUGE may fail to capture the quality of a summary, human evaluation is sometimes used, mostly in addition to automatic evaluation (Zhao et al., 2020; Wu et al., 2021). Obviously, if the goal is to evaluate a summarization system with regard to what humans deem important, employing human evaluators is desirable to cover up the flaws of automatic methods. However, using human evaluators is usually incredibly resource-demanding, both in terms of time and money, and requires extensive planning (Nenkova, 2006). Thus, human evaluation may be unfit for comparing different systems and in-development evaluation.

Zhao et al. (2020) conduct a human study by presenting participants with multiple summary pairs consisting of one human-written and one machine-generated summary. Participants are then asked to evaluate each summary according to four criteria: fluency, consistency, coverage, and redundancy (FCCR). Zhao et al. (2020) claim that in combination, FCCR covers the most important aspects of a summarization system and thus creates a solid basis for evaluation. Note that study participants in Zhao et al. (2020) have read the source documents before the summaries are presented.

### Human Trainers

Using human AI trainers to fine-tune summarization models after they have been trained on data is an interesting idea for training summarization systems further. Doing so alleviates the need to optimize an automatic metric like ROUGE, which does not necessarily align with humans' idea of a good summary (Stiennon et al., 2020). Wu et al. (2021) utilize AI trainers by implementing a reinforcement learning model which learns from human feedback. Evaluation by Wu et al. (2021) show that the model trained on human feedback was preferred compared with trained, supervised models ten times its size. This means that such approaches hold great potential for future work, and Wu et al. (2021) claim that having AI trainers is essential to make sure future summarization systems align with human intentions. However, it must be mentioned that using human feedback to train models is extremely resource-demanding, especially compared to training models solely on public datasets.

# 4. Datasets

This chapter presents three relevant datasets for the rest of the thesis. Section 4.1 describes the working dataset, which represents the task of entity-related summarization based on a collection of news articles. The working dataset is used for manual exploration and to generate summaries for the survey that was conducted. Section 4.2 and Section 4.3 presents CNN/DailyMail and Multi-News respectively: two public datasets that are used for fine-tuning in the abstractive summarization system. The test set of CNN/DailyMail is also used for automatic evaluation, which is presented in Section 6.1.

## 4.1. Working Dataset

As presented in Section 3.1, no public dataset for the specific task of entity-related multi-document summarization has been found. Thus, the primary dataset used in this master's thesis is the *working dataset*, which is explicitly made for this thesis in collaboration with Strise. It is not particularly large, but it represents the desired use case of the systems presented in this thesis: generate entity-related summaries from a collection of news articles.

The working dataset contains both collections of articles and information about entities in the articles. It was primarily used for manual exploration during the development of the summarization systems presented in Chapter 5, as well as to generate summaries for the survey that is presented in Section 6.3. It must also be mentioned that the dataset cannot be published along with the source code due to legal agreements with the providers of the news articles.

### 4.1.1. Structure

The dataset consists of two CSV files: one file with the collections of news articles and one containing the entities' location in the articles. The news article file contains ten different collections of news articles, which in the dataset are referred to as "clusters". This thesis will refer to a cluster of news articles as a "collection" to clarify whether "cluster" refers to a cluster of documents or a cluster related to the k-means algorithm (Section 2.3.1). Each collection contains up to 25 news articles related to the same topic. The topic for each collection is specifically chosen to be something people might be familiar with, which is intended for the survey and explained in greater detail in Section 6.3. The essential columns in the collection CSV file are presented in Table 4.1.

The entity file contains every entity-mention that has been found by the Strise Named-entity recognition (NER) system. The entity-mentions are described by the ID of the

| Column name | Description |
|---|---|
| collectionId | The id of the collection |
| eventId | The id of the news article |
| title | The title of the news article |
| body | The content of the news article (excluding the title) |
| fulltext | All the content of a news article (title and body combined) |

Table 4.1.: The relevant columns of the collection CSV file.

| Column name | Description |
|---|---|
| wikidataId | A unique entity identifier that is based on the Wikidata entity IDs[1] |
| collectionId | A mapping to the collection in which the entity occurs |
| eventId | A mapping to the article in which the entity occurs |
| offset | The index of the first letter of the entity in the associated article. The index refers to the *fulltext* field of an article. |

Table 4.2.: The relevant columns of the entity CSV file.

entity and a mapping to where the entity occurs. The mapping consists of the ID of the collection it is in, the ID of the news article in that collection and the index to the first letter of the entity in that article string. The entity CSV file thus has four essential columns, which are presented in Table 4.2.

### 4.1.2. Limitations

The working dataset does not include any gold standard reference summaries. Thus, there is no way of automatically checking whether the summaries generated from these documents are good. Section 4.1.1 also presented that the entity-mentions are discovered by a NER system, which means that the entity-mentions in the dataset have not been identified by humans. Consequently, some mentions might have slipped under the radar, and some mentions might have been wrongly identified. For instance, one of the collections in the dataset has the topic "Will Smith slaps Chris Rock during the Oscars", and the entity "rock music"[2] was identified by the NER system multiple times. However, manual inspection showed that those entity-mentions were wrongly identified as rock music and should indeed have been identified as Chris Rock.[3]

Most articles in the dataset have been scraped from many different websites, and thus, parts of the text may be undesirable for the summarization process. Such parts are known as *junk*. Junk sentences that appear to be common in the dataset include:

- *Ads*. A common sentence in the dataset is "Play this game for free now".

[2] https://www.wikidata.org/wiki/Q11399
[3] https://www.wikidata.org/wiki/Q4109

| Measure | Average |
|---|---|
| Unique entities per cluster | 435 |
| Mentions of the top five entities | 118 |
| Unique articles in a cluster | 21 |
| Tokens in an article | 458 |

Table 4.3.: Statistics regarding the working dataset.

- *Metadata.* Metadata is usually information about the author, the publisher, estimated reading time, or publishing date. It should be mentioned that these sentences are generally pretty short.
- *Collapsed sections.* Some articles have collapsed sections, which in the scraped document appears as an incomplete sentence that ends with a "[+]".
- *Non-English sentences.* Every article in the dataset has been labeled as English, but that does not mean parts of the texts cannot be non-English. There are, for instance, multiple sentences in Spanish across the dataset.

It should also be mentioned that some news articles are published by multiple different publishers, which means that one article might be scraped multiple times, and thus lead to duplicate articles in a collection.

### 4.1.3. Statistics

The statistics presented in Table 4.3 intend to give the reader an idea of the data in the working dataset. Notice that each collection contains 25 articles initially, but after duplicate removal, the average number of articles in a collection is 21. Thus, as the average number of tokens in an article is 458, the average number of tokens in a collection (after duplication removal) is 9618. Also, notice that the number of mentions of the five most mentioned entities in a cluster, usually relevant for an entity-related summary, exceeds a hundred on average.

## 4.2. CNN/DailyMail

The CNN/DailyMail dataset (CNN/DM) was introduced by Nallapati et al. (2016) and is widely known as one of the most versatile datasets for single-document summarization. In this thesis, the CNN/DM is used for automatic evaluation (Section 6.1) and one of the transformers used in the abstractive summarization model is fine-tuned on the dataset (Section 5.4). CNN/DM is used as the baseline dataset at NLP-progress.[4] Thus, comparing results from experimentation to state-of-the-art summarization systems is straightforward. There are two noteworthy versions of the dataset: the anonymized version and the non-anonymized version. The non-anonymized version contains the

[4] http://nlpprogress.com/english/summarization.html

| Dataset split | Number of data instances |
|---|---|
| Training | 287,113 |
| Validation | 13,368 |
| Test | 11,490 |

Table 4.4.: The standard train/validation/test split for the CNN/DailyMail dataset.

| Type | Average | Median | Shortest | Longest |
|---|---|---|---|---|
| Summaries | 51.8 tokens | 48 tokens | 9 tokens | **535 tokens** |
| Articles | 683 tokens | 613 tokens | 55 tokens | 1,954 tokens |

Table 4.5.: Statistics regarding the length of different text objects in the CNN/DailyMail test set. Punctuation is included.

original articles, while the anonymized version has replaced every named entity with a special token.

The novelty of CNN/DM was how it provided input documents of the same length as a regular news article. Datasets prior to the introduction of CNN/DM tended to focus on the task of summarizing short input sequences (one or two sentences) (Paulus et al., 2017). CNN/DM was also the first large-scale dataset with multi-sentence reference summaries. Other early datasets for abstractive summarization, like Gigaword and DUC, only had one sentence in each summary. CNN/DM is easily accessible through the Hugging Face API.[5]

### 4.2.1. Details

CNN/DM has over 300 thousand data instances in total, where each instance consists of a news article and the associated gold standard reference summary. The standard train/validation/test split can be seen in Table 4.4. As the name of the dataset suggests, the news articles are scraped from CNN and Daily Mail, which are trustworthy sources in terms of language quality. The summaries are scraped from the same sources and all are written by humans. Table 4.5 presents details regarding the length of articles and summaries. Note that numbers in Table 4.5 are calculated without removing punctuation, and thus, numbers may be slightly higher than the precise word count.

### 4.2.2. Limitations

Table 4.5 shows that the dataset has some extreme points, with the most noticeable being the longest summary of 535 tokens. Manual inspection showed that the summary with 535 tokens was mostly nonsensical, and the associated article was only 1,243 tokens. However, such instances are far and few between, which goes for summaries as short as nine tokens.

[5] https://huggingface.co/datasets/cnn_dailymail

| Type | Average | Median | Shortest | Longest |
|---|---|---|---|---|
| Summaries | 217 tokens | 219 tokens | 54 tokens | 616 tokens |
| Articles | 520 tokens | 323 tokens | **5 tokens** | **172,393 tokens** |
| Collections | 3.8 articles | 3 articles | 3 articles | 11 articles |

Table 4.6.: Statistics regarding the length of different text objects in the Multi-News test set. Punctuation is included.

Kryscinski et al. (2019) show that news articles tend to present the most important information in the first third of the article, and the articles in CNN/DM are no exception. Thus, it is normal to use *lead-3*, a text comprising the first three sentences in the article, as a baseline for comparison when calculating ROUGE scores. This is done in Section 6.2.

The summaries in CNN/DM are said to have a limited level of abstraction (Koupaee and Wang, 2018). Thus, the dataset is somewhat suitable for both extractive and abstractive summarization but not perfect for either. Chen et al. (2016) performed a human evaluation of 100 random data instances and concluded that around 25% of the samples suffer from ambiguity and coreference errors.

As one of the most widely used summarization datasets, CNN/DM has been the target of studies that explore bias in the dataset. Bordia and Bowman (2019) show some evidence of gender bias in CNN/DM, highlighting that the words "crying" and "fragile" are more associated with feminine qualities in the dataset. However, CNN/DM is the least gender-biased dataset considered in the study. The other datasets are the Penn Treebank and WikiText-2 (Bordia and Bowman, 2019).

## 4.3. Multi-News

Multi-News was introduced by Fabbri et al. (2019) and has received attention due to being the first large-scale multi-document summarization (MDS) dataset. Multi-News was made to further the research on MDS, as MDS datasets prior to Multi-News were limited to a couple of hundred data instances. One of the transformers used in the abstractive summarization system, presented in Section 5.4, is fine-tuned on Multi-News. The dataset can be downloaded through the Hugging Face API.[6]

Multi-News contains about 56 thousand data instances, but unlike CNN/DM, each instance has a collection of news articles and the associated summary. The statistics presented in Table 4.6 say that there is an average of 3.8 articles per collection, meaning there are over 200 thousand articles in total. The standard train/validation/test split for Multi-News is 44,972 training instances, 5,622 validation instances and 5622 test instances.

As with any dataset made by scraping websites, Multi-News is susceptible to junk

[6] https://huggingface.co/datasets/multi_news

data. From Table 4.6, one fascinating number is the longest article, which consists of a whopping 172,393 tokens. Manual inspection showed that this article contains an entire bitcoin transaction history, which obviously is not summary-relevant information. The shortest article of five tokens is also junk: metadata that has somehow been added as an article. As with the CNN/Daily Mail dataset, such extreme instances are rare, but it still shows that the dataset might require some data cleaning. It should also be pointed out that the median length of the articles is significantly smaller than the average length, which suggests that most of the articles are on the shorter end (less than 500 tokens). It also suggests that some articles are exceptionally long and greatly increase the average.

# 5. Architecture

This chapter aims to give a deeper understanding of the summarization systems proposed in this thesis and briefly explain the architectural choices. Section 5.1 gives a high-level overview the different parts of the system. Section 5.2 describes how the input data is processed before being passed on to the summarization models. Section 5.3 and Section 5.4 explains the structure of the extractive summarization model and the abstractive summarization model respectively. Complete implementation can be found on GitHub.[1]

## 5.1. A General Overview

Research question 1 is about checking whether removing irrelevant information regarding an entity leads to a good entity-related summary. Thus, the first phase is *data preparation*, which includes (1) cleaning the input documents, (2) dividing the documents into sentences, (3) marking each sentence with relevant information (such as entities in the sentence), and (4) removing the sentences that do not include the entity (if an entity is specified). It is important to note that making an entity-related summary is optional, and the summarization systems should be able to generate general multi-document summaries as well.

The second phase is *generating a summary*. One extractive model and one abstractive model are proposed for this phase. Thus, there are effectively two different summarization systems: both have the same data preparation phase but use different summarization models for the second phase. Consequently, the details of what happens in the second phase are dependent on the summarization model. The extractive model takes a clustering-based approach, and the abstractive model takes a transformer-based approach. In essence, the second phase turns prepared data into a summary. One thing to note is that "summarization system" refers to the entire architecture, comprising both the data preparation and the summarization (i.e., the first phase and the second phase). "Summarization model", on the other hand, refers to the part of the system that takes in prepared data and returns a summary (i.e., the second phase).

The general idea of the system's flow, from input documents to summary, is illustrated in Figure 5.1. It shows how the system begins with a collection of documents prepared in the first phase and leaves a bunch of sentence objects that the summarization model uses in the second phase to generate a summary.

Something to keep in mind throughout this chapter is one of the limiting factors for the summarization systems: no public datasets for entity-related multi-document

[1] https://github.com/martinstiles/masters-thesis

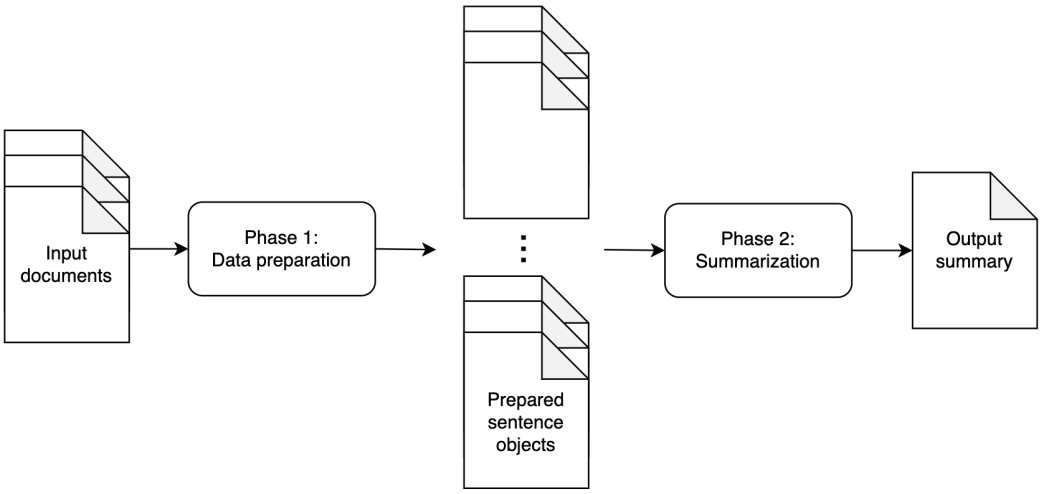



Figure 5.1.: General overview of the flow of the summarization systems.

summarization were identified during the literature review. Thus, no neural network can be specifically trained for the exact task at hand, although doing so would be a challenge. This motivates the naïve approach of removing irrelevant information with regard to an entity and then summarizing the remaining text. It is also worth mentioning that the scope for this master's thesis is a limiting factor for the architecture. Thus, some architectural choices are made based on ease of implementation.

## 5.2. Data Preparation

The data preparation phase is all about preparing the textual data for summarization. Everything specific that has to do with entities happens here. It must be mentioned that the entire data preparation phase is based on the working dataset presented in Section 4.1. For the system to work with other datasets, minor adjustments must be made, such as in Section 6.1. A simplified example of how input is processed in the data preparation phase can be seen in Appendix B.

### Identifying Entities

Identifying entities is technically a part of the data preparation phase. However, all the work is done beforehand by Strise, the company with which this thesis is written in collaboration. The entities are found using a Named Entity Recognition (NER) system. The NER system is made by Strise, and the confidentiality agreements restrict the extent to which this thesis may discuss how the system works. The NER system is based on the ideas presented by Pershina et al. (2015) and Yamada et al. (2016). Recall from Section 4.1.1 that the working dataset contains a file with entity occurrences in the news articles in the dataset. This file is essentially the output from the NER system, and it can be

used to locate sentences that contain entities of interest.

### Removing Duplicate Articles

As mentioned in Section 4.1.2, an article might occur multiple times in a collection. If this is the case, the information in that article is more likely to appear in the final summary. Duplicate sentences will be deemed more important in both the abstractive and extractive models. The transformer in the abstractive model will likely give greater attention to the duplicate sentences. During clustering in the extractive model, duplicate sentences will end up in the same cluster and thus making the cluster more likely to be chosen as the top cluster. Duplicate sentences are also more likely to be chosen as the *best sentence* because they are more semantically similar to the other sentences in the cluster. Thus, duplicate articles are undesirable and are removed at the very beginning.

### Dividing Into Sentences

Input documents must be divided into sentences before sentences that do not mention the relevant entity are removed. The articles in the dataset are already divided into smaller parts with double space as a delimiter. Thus, the articles are first split on the delimiter to eliminate the double spaces. However, after splitting the input, the smaller parts may consist of more than one sentence. Therefore, the sentence tokenizer called *sent_tokenizer* from the NLTK library[2] is applied in addition to splitting on double space.

### Constructing Sentence Objects

A sentence object, or a "dictionary" as it is called in Python, is constructed because certain values must be attached to the sentences. Note that aside from entity-mentions, attaching values to sentence objects is exclusive to the extractive model. During run-time of the extractive model, specific values like sentence embeddings must be associated with the original sentence. Thus, an object is a good way to keep all sentence-specific values in one place. The different values related to the extractive model will be presented further in Section 5.3.

### Marking Sentences With Entities

While constructing sentence objects, each sentence is marked with two offsets: the index of the first sentence symbol and the first sentence symbol. Here it is crucial to account for all the spaces that were removed while splitting the articles into sentences because the entity offset from the dataset includes the spaces. It should also be mentioned that the entity offset refers to the *fulltext* field of an article. However, the summarization systems only use the *body* field for summarization, meaning the offsets are different. Recall from

[2] https://www.nltk.org/api/nltk.tokenize.html

Table 4.1 that the *fulltext* is just the combination of *title* and *body*. Thus, the offset difference is the title's length, which is simply added to the body offset.

The IDs of the news articles are available at the time of sentence object construction. Thus, the process of marking each sentence with the entities that are in them is done in two steps. The first step is to extract the entities that are mentioned in the given news article. The second step is to iterate through every sentence in that article and mark each sentence with the entities that have an offset between the sentence's start index and end index.

### Data Cleaning

Data cleaning is all about acknowledging and dealing with the text-related limitations of the dataset, which is presented in Section 4.1.2. In this step, the goal is to remove unwanted data – primarily junk. Section 4.1.2 reveals that the junk in the news articles is mostly ads, metadata, collapsed sections, and non-English sentences. One way of dealing with junk is to use an exclude list. An exclude list is a list of terms; if a sentence contains at least one term, the sentence is excluded. Having an exclude list means that some sentences might be removed despite not being junk. However, the exclude list is quite restricted, and the most critical aspect is to reduce junk in summaries. The exclude list is made up of terms that were found to be common in junk sentences, and it is implemented as follows:

```
exclude_list = [
  "contact me",
  "play now",
  "for free",
  "getty images",
  "keep up with",
  "[+]"
]
```

Another action taken is to remove very short and very long sentences. Every sentence consisting of 8 words or less is removed. This is mainly an effort to remove the junk that the exclude list fails to capture. Metadata in the articles are typically very short, e.g., "2 min read". By default, every sentence consisting of 50 words or more is removed to exclude unnaturally long sentences. However, *max_sentence_length* is an input parameter for the systems, which is primarily used by the extractive system. Maximum sentence length is usually set to 35 in the extractive system to avoid creating very long summaries. Removing sentences with certain lengths means that some non-junk sentences might be removed, but it is assumed that there are enough remaining sentences to make a good summary.

It should be mentioned that the sentences are intentionally kept as is; without applying standard text cleaning like stemming and stop-word removal. This is because the transformers used in the summarization models are pre-trained on regular, uncleaned data. Thus, extensive text cleaning is not only unnecessary – it might even make the

transformers less effective.

#### Remove Irrelevant Sentences With Regard To Entities

Considering that each sentence is marked with entities at this point, this step is straightforward. Given a list of entities: every sentence that is not marked with at least one of the entities is removed. However, inputting one or more entities is optional, and if none is given, then no sentences are removed.

## 5.3. The Extractive Model

What is referred to as the "extractive system" comprises the data processing step and the extractive model that this section presents. The architectural design of the extractive model is primarily based on ideas from Zhao et al. (2020). Furthermore, a combination of manual exploration and identified state-of-the-art (SOTA) approaches is used to decide how to implement individual components of the model. This section aims to present the architecture of the extractive model, as well as justify the architectural choices.

### 5.3.1. General Idea

The general idea behind the extractive model can be described in four steps:

1. The sentences are encoded in a way that captures their semantic meaning.
2. Clustering is performed to group sentences of similar semantic meaning.
3. The n most important clusters are identified.
4. For each cluster of the n largest clusters, the sentence that is most semantically similar to the other sentences in the cluster is extracted.

In step 1, it is assumed that the sentence encoder manages to capture the semantic meaning of a sentence. In step 2, the hypothesis is that the clustering method groups sentences that convey similar information or have a similar topic. Step 3 assumes that the most important topics in the input collection are identified by the clusters with the most sentences (the largest clusters). Finally, step 4 assumes that the model manages to find the most representative sentence in a cluster, and by extracting the top sentence from the n largest clusters, the sentences should be about different topics. Thus, the end hypothesis is that the resulting summary consists of n *best* sentences that represent the n *most important* topics throughout the documents and has minimum repetition.

### 5.3.2. Sentence Encoding

Since a clustering approach is used, the sentences must be transformed into a sentence representation suitable for clustering. There are many ways to represent sentences, but the current state of the art uses pre-trained transformers. Compared to classical

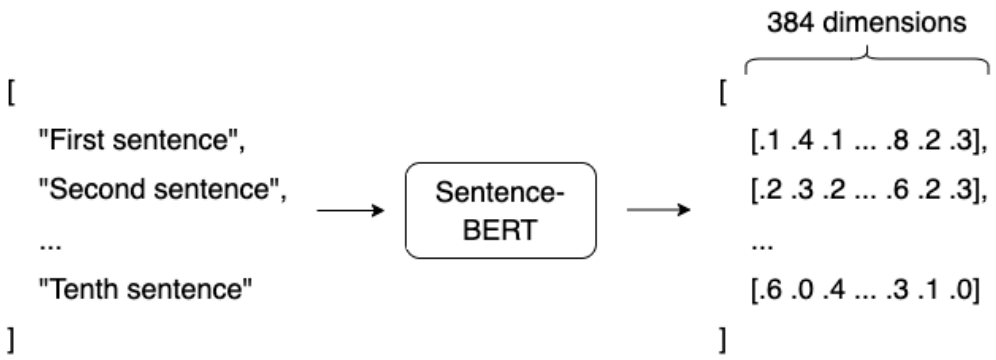



Figure 5.2.: Sentence encoding illustration.

sentence representations like bag-of-words (Section 2.2) or Sent2Vec (Pagliardini et al., 2018), pre-trained sentence-transformers are vastly superior with regard to capturing the semantic meaning of sentences. This is crucial as the premise for the clustering approach is that the semantic meaning of sentences is captured in the sentence representations. However, the ability to capture semantic meaning well comes at a cost, as sentence transformers are relatively slow.

The sentence transformer used in this step is Sentence-BERT (Reimers and Gurevych, 2019). Sentence-BERT is chosen due to the ease of use with the Hugging Face API,[3] not to mention that the sentence-transformer has achieved excellent results in various natural language processing tasks (Li et al., 2020). Sentence-BERT transforms any sentence into a 384-dimensional dense vector space. The vector representing a sentence is called a "sentence embedding". Additionally, a mini version of Sentence-BERT is used to combat the issue of inefficiency. Thus, the sentence encoding process is relatively fast and yields good sentence embeddings.

The sentence encoding process is illustrated in Figure 5.2. Of course, the illustration in Figure 5.2 is an oversimplification, but it shows how sentences are transformed into embeddings. For each remaining sentence object from the data preparation phase (Section 5.2), the sentence embedding is calculated using Sentence-BERT and then attached to the object.

### 5.3.3. Clustering Sentences

At this point, every sentence object has been labeled with its sentence embedding. The sentence embeddings are 384-dimensional vectors representing the original sentences' semantic meaning. Thus, the purpose of clustering is to create groups of semantically similar sentences. Continuing the example in Figure 5.2, Figure 5.3 illustrates how the ten sentences might be clustered based on their sentence embeddings. Every "X" represents the embedding of one of the ten sentences and the red circles illustrates how the sentences

[3] https://huggingface.co/sentence-transformers/paraphrase-MiniLM-L6-v2

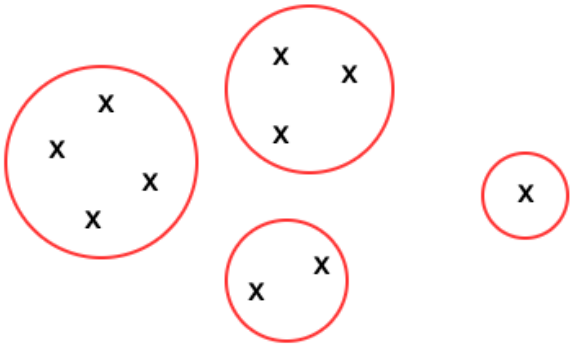


Figure 5.3.: Clustering illustration, continuing the example from Figure 5.2.

have been grouped. The example in Figure 5.3 is illustrated in two-dimensional space for visual purposes, but in reality, the embeddings exist in a 384-dimensional space.

### K-Means for Sentence Clustering

The performance of a clustering algorithm is limited to the quality of the data points that should be clustered. Thus, it is assumed that as long as the sentence embeddings are good, the choice of clustering algorithm will not significantly impact results. Consequently, the clustering algorithm's primary criteria are efficiency and ease of implementation. Here, K-means is a great candidate, as it is widely used due to its efficiency and straightforward implementation (M. Salih and Jacksi, 2020). The K-means algorithm is described in more detail in Section 2.3.1.

The *KMeans* class provided by the Python library scikit-learn[4] is used. The implementation of K-means is heavily inspired by Arvai et al. (2020). The maximum number of iterations is set to 150 to prevent the algorithm from accidentally entering an infinite loop, and the initial centroids are given random values.

### Selecting K in K-means

The number of clusters (K) must be predetermined when using K-means. This is a problem, as there is no way of knowing which value of K will lead to the best summary beforehand. However, certain measures can be taken to estimate the best value of K, and one of them is to calculate *silhouette coefficients*. A silhouette coefficient, presented in detail in Section 2.3.1, is a measure of how well a value of K is able to cluster the data points. Thus, the silhouette coefficient is measured for a range of K values, and the K with the highest silhouette coefficient is chosen. In the extractive model, the silhouette coefficient is calculated for values of K ranging from 5 to 12. The range is small for efficiency purposes, but manual exploration also suggested that values for k larger than

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

12 resulted in very incoherent summaries. Note that the implementation of silhouette coefficient is also inspired by Arvai et al. (2020).

### 5.3.4. Choosing Sentences To Extract

The extractive model has an input parameter called *num_sentences*, which describes the number of sentences to be extracted and included in the final summary. *num_sentences* will be referred to as $n$ in this section. The general idea of choosing sentences is to (1) identify the *best* n clusters, (2) identify the *best* sentences in the *best* n clusters, and then (3) combine those sentences in a chronological order with regard to their position in their original articles. Figure 5.4 illustrates this process. Continuing the example from Figure 5.3, the green circles in Figure 5.4 shows the *best* clusters (note that $n = 2$). The *best* sentence in both of the two clusters are extracted and concatenated in the final summary.

#### Identifying the *Best* n Clusters

"The *best* n clusters" refers to the n clusters representing the most important topics in the source documents. As presented in Section 5.3.3, clustering aims to group semantically similar sentences. Assuming that this is achieved through K-means, each cluster will represent different topics in the source documents. Thus, by extracting one sentence from different clusters, the resulting summary should have little repetition across sentences and good coverage of important topics in the source documents. Minimizing repetition is crucial, as redundant information is one of the most common problems for extractive multi-document summarization (Calvo et al., 2018). However, minimizing repetition in such a way might come at the cost of coherence.

The n largest clusters (clusters with most sentences) are regarded as the *best*. If a cluster has many semantically similar sentences, it should imply that the type of sentences in that cluster is important to the source documents. Thus, the process of identifying the best clusters is to find the cluster with the most sentences.

#### Extracting the *Best* Sentence in the Top-n Clusters

The *best* sentence in a cluster is the sentence that is most semantically similar to the other sentences in that cluster. The approach to finding the best sentence is inspired by the TextRank algorithm (Section 2.2): for each cluster in the n best clusters, calculate the similarity between every pair of sentences in the cluster using their sentence embeddings. Cosine similarity (Section 2.2) is used to measure semantic similarity between two sentences, as it is one of the most popular document similarity measures (M. Salih and Jacksi, 2020). Thus, the sentence with the lowest total cosine similarity score in each cluster is extracted.

It is worth mentioning that, at first, a TF-IDF measure (Sammut and Webb, 2010) for each sentence was used to find the most representative sentence in a cluster. However, manual exploration suggested that, with this measure, the chosen sentences were often poor summary sentences. The TF-IDF approach was also biased towards extracting

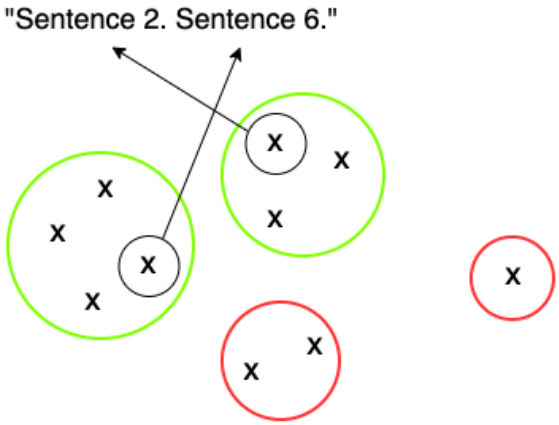



Figure 5.4.: Extraction illustration, continuing the example from Figure 5.3.

long sentences, making the final summaries lengthy. A punishment for long sentences was introduced to combat this bias. However, that caused gibberish sentences to be extracted. In hindsight, the TF-IDF approach is regarded as unfit for finding the most representative sentences, as the TF-IDF measure rewards uniqueness in sentences.

### 5.3.5. Structuring and Concatenating Extracted Sentences

At this point, the n *best* sentences have been extracted. The final step is to concatenate the sentences sensibly and return the summary. A measure of a sentence's location in its associated article is attached to every sentence object during data preparation. For instance, if a sentence is the second sentence of a ten-sentence long article, the measure is 2/10. Thus, the extracted sentences are sorted in ascending order based on sentence location measures. This is why "Sentence 2" comes before "Sentence 6" in Figure 5.4, because that is the sentence ordering in the original document. The sentences are then concatenated to form the final summary. It must be mentioned that this method does not guarantee a sensible ordering of sentences when there are multiple source documents. However, manual experimentation suggested that it improved coherence in most cases.

## 5.4. The Abstractive Model

What is referred to as the "abstractive system" comprises the data processing step and the abstractive model that this section presents. In contrast to the extractive model, the abstractive generates entirely new sentences for the summary. This is where pre-trained transformers come into play, as they are exceptional at language generation and can be fine-tuned for summarization tasks. This section presents how pre-trained transformers are used in this thesis to summarize potentially large amounts of texts.

### 5.4.1. General Idea

The idea behind the abstractive model is to take a SOTA pre-trained transformer for abstractive summarization, and make it work in a multi-document environment. A limitation of pre-trained transformers is that they have a maximum input size. Thus, the idea is to combine sentences from the data preparation phase into chunks such that each chunk fits into the transformer model. Consequently, each chunk can be summarized by the transformer model. If the length of the combined summaries is longer than the desired summary length, then the process is repeated. Otherwise, the summary is returned. Thus, the approach can be said to be a *recursive approach*.

### 5.4.2. Chunking

As mentioned in Section 5.4.1, chunking combines the input sentences so that a SOTA pre-trained transformer can summarize different parts individually. Pre-trained transformers for abstractive summarization usually have a maximum input size of 512 tokens. Thus, a chunk can contain a maximum of 512 tokens. Chunks are iteratively made by adding sentences to a chunk until the point where adding the next sentence to the chunk will cause the chunk to exceed 512 tokens. At that point, a new chunk is made. Finally, each chunk is represented by combining the sentences in the chunk into one string.

### 5.4.3. Summarizing Chunks

The following paragraphs presents why the PEGASUS model (Zhang et al., 2019) was chosen and how it is used to summarize chunks.

#### Choosing Transformer Model

The initial plan was to find a SOTA pre-trained transformer, e.g., BART (Lewis et al., 2019) or PEGASUS (Zhang et al., 2019), and fine-tune it on an abstractive summarization dataset with news articles. When fine-tuning a transformer model, it is preferable to use a dataset similar to the intended use case because the model may learn to recognize specific patterns for that kind of data. The intended use case for the summarization models in this thesis is to summarize collections of news articles. Thus, the two datasets chosen for fine-tuning are CNN/Daily Mail (CNN/DM) and Multi-News, which are presented in detail in Section 4.2 and Section 4.3 respectively.

It was discovered that the Hugging Face API provides two PEGASUS models that are fine-tuned on CNN/DM[5] and Multi-News[6] respectively. PEGASUS, as presented in Section 2.3.2, is a transformer model that has achieved great results for abstractive summarization tasks. Thus, for the sake of convenience, both of the fine-tuned transformer models are downloaded via the Hugging Face API by using the *transformers* library.

[5] https://huggingface.co/google/pegasus-cnn_dailymail
[6] https://huggingface.co/google/pegasus-multi_news

### Using PEGASUS

Two PEGASUS models (PEGASUS CNN/DM and PEGASUS Multi-News) are downloaded. Note that only one of the PEGASUS models is used at one time. The abstractive system has an input parameter called *model_type* – a string specifying which transformer model to use (PEGASUS CNN/DM or PEGASUS Multi-News). This effectively means that there are two abstractive systems: one that uses PEGASUS CNN/DM and one that uses PEGASUS Multi-News. When discussing different models later in this thesis, the systems are referred to as "Abstractive CNN/DM" and "Abstractive Multi-News", depending on which PEGASUS model is used. However, aside from the PEGASUS model used, the abstractive systems are identical.

A *summarization pipeline*[7] is created to use the models for summarization. A summarization model and a tokenizer should be specified when initiating the summarization pipeline. The summarization model is either of the PEGASUS models, depending on which one should be used. The tokenizer is a pre-trained tokenizer associated with the PEGASUS model that is used, which can be accessed through the Hugging Face API.[8] When the pipeline is initiated, summarizing is as easy as calling the *pipeline.summarize()* method.

The *pipeline.summarize()* method is called with three input parameters to summarize a chunk: the chunk (a string consisting of fewer than 512 tokens), *min_length* (the minimum desired number of tokens in the generated summary), and *max_length* (the maximum desired number of tokens in the generated summary). By default. *min_length* and *max_length* is set to 50 and 80 respectively.

### Removing Unfinished Trailing Sentences

The last sentence of the generated summary tends to be incomplete. This mostly seems to happen when the transformer model reaches the maximum length and stops generating new words, which leaves the last sentence unfinished. Thus, if the last sentence of a generated summary does not end with punctuation, it is removed.

It must be noted that this fix makes it possible to generate summaries that are shorter than the minimum desired length. However, this does not seem to happen too often, and in cases where it does happen, the length of generated summaries tends to be slightly below the minimum length. Therefore, it is not considered a significant problem – but something to be aware of.

## 5.4.4. Recursive Approach

The recursive approach used in the abstractive model is heavily inspired by the recursive summarization approach presented by Sinha et al. (2018). The approach is illustrated in Figure 5.5. The steps in Figure 5.5 are as follows:

[7] https://huggingface.co/docs/transformers/v4.19.2/en/main_classes/pipelines#transformers.SummarizationPipeline

[8] https://huggingface.co/docs/transformers/main_classes/tokenizer

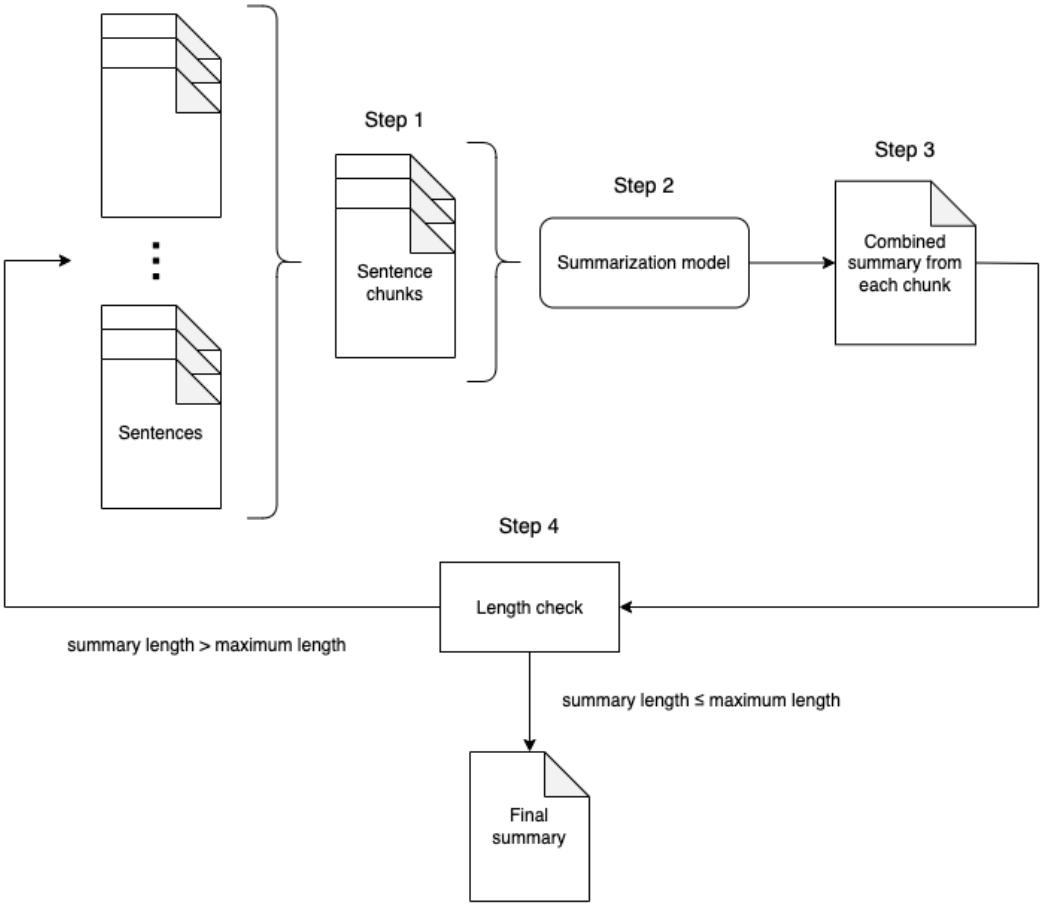



Figure 5.5.: Illustration of the recursive approach.

1. Divide the input into chunks (described in Section 5.4.2).
2. Generate a summary for each chunk (described in Section 5.4.3).
3. Concatenate every generated summary from the previous step and let the concatenation represent the *new summary*.
4. Expose the new summary to a length check: if the number of tokens in the new summary exceeds the maximum desired number of tokens, the process is repeated from step 1 with the new summary as input. However, if the number of tokens is adequate, the summary is returned.

The abstractive system has an input parameter called *max_summary_length*, which is set to 80 tokens by default. Thus, the maximum desired number of tokens in a summary is 80 by default. There is also an input parameter for the minimum desired number of tokens called *min_summary_length*, which is set to 50 by default. However, this restriction is only enforced by the transformer model. In the end, generated summaries contain between 50 and 80 tokens by default but can easily be changed.

# 6. Experiments and Results

Evaluating the summarization systems is a vital part of the thesis. Therefore, thorough experiments must be carried out to create a basis for evaluation. In this thesis, two types of experiments are conducted: *automatic evaluation* and *human evaluation*. Automatic evaluation refers to using a labeled dataset and calculating certain metrics that represent the model's performance. Human evaluation refers to the act of making humans judge aspects of the system, which in this thesis is done by using a survey. In combination, automatic and human evaluation can complement each other's limitations well. However, the conducted experiments still have fundamental flaws, which are discussed in more detail in Section 7.2.1 and Section 7.2.2. The following sections present how the experiments were conducted and the relevant results.

## 6.1. Automatic Evaluation Plan

Automatic evaluation is a low-effort method to measure the performance of a summarization model. In this experiment, a dataset is used to measure the performance of the different models. However, the performance measures do not say much by themselves and are primarily used to compare different models.

This section describes in detail how automatic evaluation is performed, despite a lack of relevant datasets. It also discusses why different choices were made, especially concerning the limitations of the chosen experimental approach.

### 6.1.1. Dataset

No public datasets for entity-related summarization have been identified, as discussed in Section 3.1. Consequently, it is not possible to automatically evaluate the summarization models from this thesis regarding their ability to generate entity-related summaries. The main purpose of the automatic evaluation is thus to measure the models' ability to compress textual information. For this purpose, the CNN/DailyMail dataset (CNN/DM) was chosen.

#### Specific Details

CNN/DM, presented in more detail in Section 4.2, is a single-document abstractive summarization dataset. It is widely used for summarization tasks, which makes it optimal for comparing the models in this thesis with state-of-the-art (SOTA) summarization models. The dataset is downloaded via the Hugging Face API by using the *datasets*

Python library. The experiment is conducted on the test split provided by Hugging Face, consisting of 11,490 data instances.

#### Addressing Limitations Regarding the Dataset

The reference summaries in CNN/DM are abstractive, suggesting that abstractive models should have an advantage when using the dataset for evaluation. However, as Section 4.2.2 mentions, the summaries in CNN/DM have a limited level of abstraction. Thus, extractive models are not at a serious disadvantage. In fact, many extractive models achieves relatively high ROUGE scores on CNN/DM, as presented on NLP-progress.[1]

CNN/DM is a single-document summarization (SDS) dataset, The models presented in this thesis are not designed for single-document summarization (SDS), and CNN/DM is a SDS dataset. However, this experiment aims to evaluate the models' ability to compress information – and CNN/DM is arguably the best dataset for that purpose. Since the design of the summarization models in this thesis are not intended for single documents, the most important findings will be to compare their results. The results will also be compared to SOTA summarization models, but mostly to put things in perspective.

One question remains: *is it reasonable to use CNN/DM for evaluation when one of the abstractive models uses a transformer that is fine-tuned on the same dataset?* Zhang et al. (2019) mention that the datasets used for fine-tuning are accessed through TensorFlow Summarization Datasets[2], and the provided train/validation/test split is used. The CNN/DM dataset provided by TensorFlow[3] has the same train/validation/test split as the CNN/DM dataset provided by Hugging Face, which is where this thesis accessed the dataset. Thus, PEGASUS CNN/DM has not been trained on the test instances of CNN/DM, and it is fair to use the dataset for evaluation. However, it should be noted that PEGASUS CNN/DM still has an advantage over PEGASUS Multi-News, as it has been trained on data similar to the evaluation dataset.

### 6.1.2. Evaluation Metrics

ROUGE metrics are the de facto standard for evaluating summarization models. ROUGE-1, ROUGE-2, and ROUGE-L are the most widely used ROUGE metrics and constitute the three metrics used for comparison on NLP-progress. Thus, to easily compare with other summarization models, ROUGE-1, ROUGE-2, and ROUGE-L scores are calculated. The Python library *rouge-score* is used to calculate all ROUGE scores. The ROUGE scores associated with a summarization model are the average of the scores the model got for each data instance of the dataset.

[1] http://nlpprogress.com/english/summarization.html
[2] https://www.tensorflow.org/datasets/catalog/overview
[3] https://www.tensorflow.org/datasets/catalog/cnn_dailymail

### 6.1.3. Model Parameters

Only the parameters regarding the length of generated summaries are specifically chosen for this experiment. The mean number of tokens in the reference summaries in CNN/DM is 56, and thus, generated summaries should also be around that length.

The relevant parameters for the extractive model are *num_sentences*, the number of sentences to extract, and *maximum_sentence_length*, the maximum number of words extracted sentences can contain. Two different sets of parameters are tested for the extractive model. Both sets have $num\_sentences = 2$, but the first set has no upper bound on sentence lengths, while the second set has a maximum length of 35 words. Thus, two slightly different versions of the extractive model are tested. These versions are briefly presented in Section 6.2.

The transformers in the abstractive model are time-consuming. Consequently, only the default values are tested. Thus, the minimum number of tokens is 50, and the maximum is 80.

### 6.1.4. Other Models for Comparison

As mentioned, the most valuable aspect of this experiment is to compare the thesis models to each other. However, it is also interesting to see how the thesis models compare to SOTA summarization models. Additionally, it is common to use lead-3 as a baseline for CNN/DM, as news articles tend to have the most important information in the first third of the article (Kryscinski et al., 2019). Thus, the models that are used for comparison are as follows:

**Lead-3** As discussed in Section 4.2, news articles tend to have the most important (and thus summary-relevant) information in the first third of the article. Lead-3 is simply the concatenation of the first three sentences of an article.

**PEGASUS** The abstractive model in this thesis uses PEGASUS, either fine-tuned on CNN/DM or Multi-News, but with a chunking approach for the input document(s). Thus, the results from the standard implementation of PEGASUS (Zhang et al., 2019) are included for comparison. Note that the ROUGE scores for CNN/DM in Zhang et al. (2019) are from the model that is fine-tuned on CNN/DM.

**MatchSum** The best performing extractive model on NLP-progres, proposed by Zhong et al. (2020).

**SimCLS** The best performing abstractive model on NLP-progres, proposed by Liu and Liu (2021).

## 6.2. Automatic Evaluation Results

This section presents the results from the automatic evaluation. First, the results from the models presented in this thesis are presented and compared to each other. Secondly,

| Metric | Extractive (none) | Extractive (35) | Abstractive CNN/DM | Abstractive MN |
|---|---|---|---|---|
| ROUGE-1 P | **29.46** | **31.07** | 43.13 | 32.74 |
| ROUGE-1 R | **37.11** | **34.67** | 41.54 | 35.03 |
| ROUGE-1 F1 (mean/std) | 31.51/11.35 | **31.65**/11.52 | **40.69**/13.14 | **32.50**/10.66 |
| ROUGE-2 P | 9.60 | 10.11 | 20.05 | 10.42 |
| ROUGE-2 R | 11.82 | 11.10 | 19.13 | 11.20 |
| ROUGE-2 F1 (mean/std) | 10.15/9.99 | **10.20**/**10.13** | **18.81**/13.59 | **10.34**/8.21 |
| ROUGE-L P | 17.88 | 19.10 | 30.33 | 19.90 |
| ROUGE-L R | 22.64 | 21.47 | 29.31 | 21.54 |
| ROUGE-L F1 (mean/std) | 19.15/9.00 | 19.51/9.25 | 28.66/12.84 | 19.84/7.63 |

Table 6.1.: All ROUGE scores for the summarization models in this thesis. Results of special interest are emboldened.

models from external sources are considered to see how the models in this thesis compare to state of the art.

### 6.2.1. Results for the Thesis Models

ROUGE scores based on the CNN/DM test set have been calculated for the following four versions of the thesis models:

**Extractive (none)** The extractive model with no maximum sentence length.

**Extractive (35)** The extractive model with a maximum sentence length of 35 words.

**Abstractive CNN/DM** The abstractive model using PEGASUS CNN/DM.

**Abstractive MN** The abstractive model using PEGASUS Multi-News.

Table 6.1 presents the results for each of the models above. Precision (P), recall (R), and F1 scores for each ROUGE metric are included. The most important score is the F1 score, as it captures the harmonic mean of precision and recall. Thus, for every F1 score, both the mean and the standard deviation are presented. However, precision and recall scores are also included to further the basis for discussion.

The first thing to notice is that Extractive (none) does indeed get higher recall scores and lower precision scores compared to Extractive (35). This is expected, as the summaries with no maximum limit are longer on average. Recall from Section 2.4.1 that precision and recall are impacted differently by the summary length. However, the F1 scores for Extractive (none) and Extractive (35) are almost identical, suggesting that sentence

| Model | ROUGE-1 | ROUGE-2 | ROUGE-L | Source |
|---|---|---|---|---|
| Extractive (35) | 31.65 | 10.20 | 19.51 | Thesis (Section 5.3) |
| Abstractive CNN/DM | **40.69** | 18.81 | 28.66 | Thesis (Section 5.4) |
| Abstractive MN | **32.50** | 10.34 | 19.84 | Thesis (Section 5.4) |
| Lead-3 | 40.05 | 17.48 | **25.02** | Thesis |
| Lead-3 | 40.34 | 17.70 | **36.57** | See et al. (2017) |
| PEGASUS | **44.17** | 21.47 | 41.11 | Zhang et al. (2019) |
| MatchSum | 44.41 | 20.86 | 40.55 | Zhong et al. (2020) |
| SimCLS | 46.67 | 22.15 | 43.54 | Liu and Liu (2021) |

Table 6.2.: ROUGE (F1) scores from models presented in this thesis and SOTA models from NLP-progress. Results of special interest are emboldened.

lengths do not significantly impact the scores. Overall, the scores for Extractive (35) turns out to be better, although very slightly.

Table 6.1 shows that Abstractive CNN/DM greatly outperforms Abstractive MN for all ROUGE metrics. Most notable is the difference between ROUGE-2 (F1) scores, where Abstractive CNN/DM has almost double the score of Abstractive MN. Additionally, precision and recall scores for Abstractive CNN/DM are very similar, while recall scores for Abstractive MN are much higher than the precision scores. This suggests that Abstractive MN generates longer summaries on average than Abstractive CNN/DM.

When comparing every model in Table 6.1, Abstractive CNN/DM clearly comes out on top for every version of ROUGE. However, it is interesting that the extractive models are almost on par with Abstractive MN, which is promising for the extractive system design.

Another thing to note is that every model has relatively high standard deviation (std) compared to the mean for every version of ROUGE. Especially notable are the ROUGE-2 (F1) scores for the extractive models, which have almost identical standard deviation and mean. High standard deviation implies that the performance of the models varies greatly between different data instances in the CNN/DM dataset.

### 6.2.2. Results in Comparison With State of the Art

Table 6.2 presents ROUGE-1 (F1), ROUGE-2 (F1) and ROUGE-L (F1) for the thesis models and the best scoring state-of-the-art models from NLP-progress, as presented in Section 6.1.4. In this section, only Extractive (35) is considered of the extractive versions for simplicity.

Table 6.2 shows that the models presented in this thesis score much lower than the SOTA models. However, one must remember that Extractive (35) and Abstractive MN are not trained on the CNN/DM dataset and are thus at a disadvantage. Abstractive

CNN/DM is the only thesis model that almost holds up, but still, it scores substantially lower than PEGASUS, which is the exact same model as Abstractive CNN/DM uses but without chunking. This suggests that chunking is not favorable for SDS. However, chunking was never intended for SDS but multi-document summarization.

An interesting observation is that the lead-3 baseline outperforms both Extractive (35) and Abstractive MN. Consequently, according to ROUGE measures, extracting the first three sentences is better than using the extractive model or Abstractive MN for the CNN/DM dataset.

The keen reader might also notice that the difference between the ROUGE-1 score and the ROUGE-L score in the thesis models is much greater than in the SOTA models. Relatively speaking, the ROUGE-L score seems to be much lower in the thesis models. This is even more apparent when comparing the lead-3 from this thesis and the lead-3 from See et al. (2017). The two lead-3 models have almost identical ROUGE-1 and ROUGE-2 scores, but the ROUGE-L score from this thesis is much lower ($25.02 < 36.57$). This is essentially proof that the method used to calculate ROUGE scores in this thesis, which is from the *rouge-score* Python library, does something different when calculating ROUGE-L.

## 6.3. Survey Design

Getting data on what humans think about the summarization systems is desirable. For that purpose, a survey was conducted, which presented examples of generated entity summaries and asked respondents to evaluate them according to certain aspects of the summary. Additionally, some questions regarding the respondents' demographics were asked. The entire survey can be found in Appendix C. Note that the survey was conducted in Norwegian, but details presented in this section have been translated to English for consistency.

The survey is divided into five pages. The first four pages have questions regarding generated entity summaries. The summaries presented in the first four pages are based on four collections of news articles from the working dataset. Each page represents one of the collections and presents the following to the respondents: (1) a short sentence about the topic of the collection, (2) the entity that the following summaries are related to, and (3) two entity-related summaries based on the collection (one extractive and one abstractive). The final page of the survey consists of demographic questions.

### 6.3.1. Generating Example Summaries for the Survey

As Section 4.1 mentions, each collection in the working dataset contains news articles related to one specific topic. These topics are specifically chosen with the survey in mind. The best-case scenario would be to ask the respondents if they think a summary is a *good summary* with regard to the documents it is based on. However, to be able to ask such a question, it is required that the respondents have read the source documents. Otherwise, it is impossible to tell whether the summary is good. Considering that each collection

| Topic of collection | Relevant entity |
|---|---|
| Johnny Depp and Amber Heard defamation trial | Johnny Depp |
| Rust shooting incident: a fatal on-set shooting where Alec Baldwin shot cinematographer Halyna Hutchins with a presumably unloaded prop gun | Alec Baldwin |
| Will Smith slapping Chris Rock on-stage during the Oscars 2022 | Will Smith |
| Rumors of soccer player Erling Braut Haaland switching teams | Erling Braut Haaland |

Table 6.3.: The chosen topics and entities for summaries in the survey.

in the working dataset contains up to 25 news articles, it is completely infeasible to make the respondents read the source documents on which the summaries are based – at least for a survey of this kind. Thus, to give respondents knowledge about the source documents without reading them, each collection in the working dataset has a topic that people are likely to be familiar with. Examples of such topics are Elon Musk buying Twitter and Will Smith slapping Chris Rock at the Oscars.

Before the survey was made public, the survey was tested by a test-respondent. Feedback suggested that four was the optimal number of topics. The test-respondent claimed that four topics were just enough, and if there were more, respondents would likely grow tired and lose interest while taking the survey.

The working dataset has twelve collections in total, but four had to be chosen for the survey. The four chosen collections are the collections that contain news articles that best reflect what people would expect from the topic of the collection. Again, this is done to increase the respondents' familiarity with the source documents. Note that in a real-world use case, one would not necessarily know the topic of the source documents beforehand, and presenting the topic in the survey is only done to allow people to evaluate the summarization models' ability to remain on-topic. The topics of the four chosen collections and the relevant entities are presented in Table 6.3.

At last, it should be mentioned that all eight summaries (four extractive and four abstractive) included in the survey were generated using the default parameters of the summarization systems. For the extractive system, a summary consists of 3 sentences with a maximum of 35 words. By default, the abstractive summaries have a minimum length of 50 tokens and a maximum length of 80 tokens. For the abstractive system, an additional choice had to be made: use PEGASUS CNN/DM or PEGASUS Multi-News. At the time, no metrics indicated which model was best at generating entity-related multi-document summaries, and manual exploration indicated that the abstractive system produced sensible summaries using either model. Thus, PEGASUS CNN/DM was arbitrarily chosen and used to generate the four abstractive summaries for consistency. Consequently, PEGASUS Multi-News is not tested in the survey.

### 6.3.2. Questions Regarding Summaries

As mentioned, the four pages with questions regarding summaries present the respondents with the topic of the source documents, the relevant entity, and two entity-related summaries (one extractive and one abstractive). These pages contain of the same questions, but the questions are related to different topics and summaries.

The first question is *Are you familiar with the topic?*, and may be answered with "Yes", "No" or "Somewhat". The question is asked to check if the respondents are familiar with the topics, which was the goal when choosing collections, as discussed in Section 6.3.1.

Next up are the questions regarding the quality of the summaries. For both the extractive and the abstractive summary, the respondents must answer to what degree they agree with the five statements. The respondents have four answer options for each statement: "Disagree", "Somewhat disagree", "Somewhat agree" and "Agree". The five statements are as follows:

1. The language is fluent, and the sentences fit together
2. The summary has little repetition
3. It is clear that the summary is about the topic
4. It is clear that the summary is about the entity
5. The summary could have been written by a human

Statements 1 and 2 are based on the FCCR measures presented in Section 3.2.4. Statement 1 is designed around the F (fluency) and measures the language quality of sentences and the coherence. Statement 2 is designed around the R (redundancy) and measures repetition. The Cs (consistency and coverage) are technically impossible to make questions about without making the respondents read the source documents first, and thus no questions are designed around the Cs. Statements 3 and 4 are tied to the summarization systems' ability to stay on-topic and whether the summary is entity-related or not. Statement 5 is to see how the respondents compare the summaries to something of human quality, but it is mostly tied to the quality of the language.

For each presented topic and the two associated summaries, the respondents are asked *Which summary do you prefer?*, similar to what Zhao et al. (2020) did. The respondents are not told whether the summary is extractive or abstractive. Thus, this question can be used to identify which summarization system people prefer more directly. Of course, it should also be possible to identify which system people prefer by interpreting the answers to the statements above.

The final question of the entire survey, which is on the fifth page but still related to the summaries, is *Could you imagine using a summarization system of this kind (for any purpose)?* The possible answers to this question are "Yes", "Yes, if it worked better", "No" and 'Ì don't know". The responses to this question can, for instance, support the motivation for future work within the field of automatic summarization.

### 6.3.3. Questions Regarding Demographics

The last page of the survey has questions regarding the respondents' demographics. The questions, along with the answer options, are as follows:

1. Question: What is you gender?
   Options: "Female", "Male", "Other"

2. Question: How old are you?
   Options: "Under 18 years", "18-30 years", "Above 30 years"

3. Question: Do you normally work with text or language?
   Options: "Yes", "No"

4. Question: How would you rate your English skills?
   Options: A scale from 1 to 6

The first two questions are asked to get a general idea of the demographics, with the goal of checking the diversity in the group. Having at least some variety among the respondents is desirable.

The third and fourth question intends to evaluate the respondents' ability to evaluate summaries. Evaluating certain aspects of a summary should not be arduous for a human. Thus there are no prerequisites for responding to the survey except basic English skills. However, it is preferable that respondents are skilled in English (with regard to questions about language quality) and deal with text/language on a professional basis (with regard to questions about summary quality). Therefore, the third and the fourth question asks about these aspects.

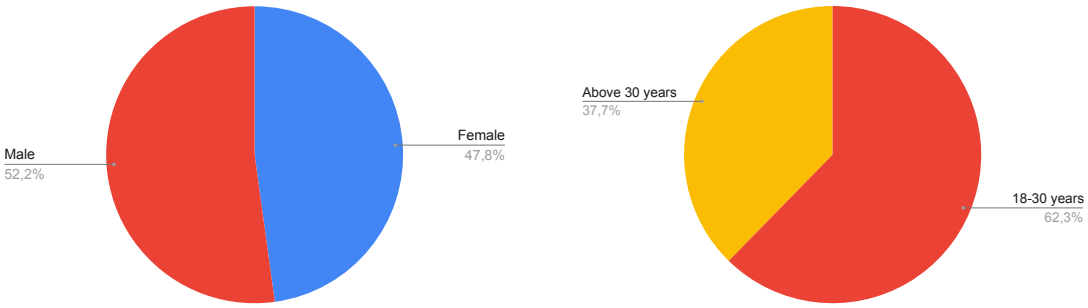
## 6.4. Survey Results

The survey had a total of 69 respondents. Note that every question was marked as mandatory, and thus every respondent has answered every question. This section will present the empirical data gathered from the survey and comment on trends and indications.

### 6.4.1. Demographics

The survey was sent out to classmates, close friends, family and close relatives, and employees at Strise. It was also advertised on the author's Facebook page. Thus, it is essential to note that most respondents are connected to the author. Furthermore, many who responded to the survey reported that they forwarded it to their close connections, giving the survey a slightly wider audience.

Recall the four questions regarding demographics presented Section 6.3.3. Figure 6.1 presents the answers to the first and the second question. Figure 6.1a shows the gender distribution of the respondents, and it is very close to a 50-50 split. This is great,




(a) Gender distribution (red: male, blue: female)

(b) Age distribution (yellow: Above 30 years, red: 18-30 years)

Figure 6.1.: Gender and age distribution in the survey.

considering gender diversity is desirable. Figure 6.1b shows the age distribution. None of the respondents belong to the group "under 18 years", but it makes no difference. What is interesting to see is the distribution between the groups "18-30 years" and "over 30 years". Respondents between 18 and 30 are regarded as students or young workers, and Figure 6.1b shows that most respondents belong to this group. Knowing that most respondents are between 18 and 30 years, and taking the people that the survey was advertised to into consideration, it is very likely that a substantial part of the respondents is technology students or people who work closely with technology. Again, based on the people the survey was advertised to, the respondents over 30 years are less likely to work with technology, but this age group is not as substantial as 18-30 years.

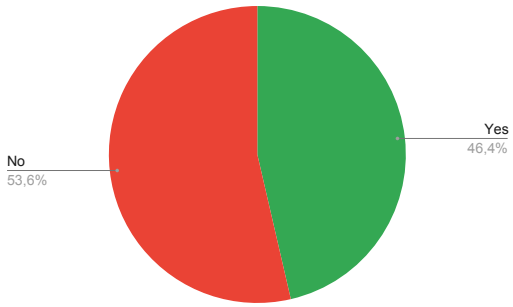
Figure 6.2 presents the results from the third and the fourth question regarding demographics. Figure 6.2a shows that almost half of the respondents regularly work with text and language. In addition, figure 6.2b shows that the respondents are reasonably skilled in English, with the average respondent rating their English skill as a 4.8 on a scale from 1 to 6. These results support the claim that the respondents are well-suited to evaluate the generated summaries, at least with regard to language quality.

### 6.4.2. Results Regarding the Summaries

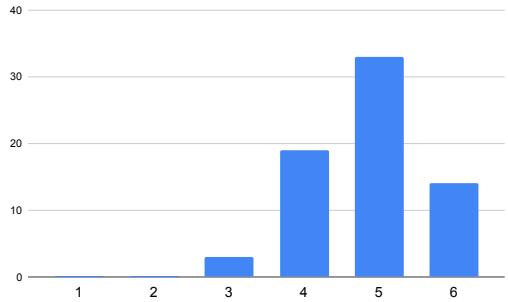
Results concerning the summaries are presented in parts such that each part reflects either the statements or one of the questions.

#### Statements

Recall the five statements presented in 6.3.2, each related to one aspect of the generated summaries. For all eight summaries, the respondents must say if they disagree, somewhat disagree, somewhat agree, or agree with each statement. Four of the summaries have been generated from the extractive system (Section 5.3) and the other four by the abstractive system (Section 5.4). The results regarding the statements will be used to compare the




(a) Respondents working with text/language (red: No, green: Yes)




(b) English skills distribution (scale from 1 to 6)

Figure 6.2.: Distribution of answers to questions regarding respondents' ability to evaluate summaries.

systems and evaluate their performance individually. Therefore, the data associated with the two systems are separated. Figure 6.3 and Figure 6.4 present the average of each response to each statement for the extractive summaries and the abstractive summaries respectively.

As Figure 6.3 and Figure 6.4 show, both systems have most of the responses in "Somewhat agree" or "Agree" for almost every statement. This suggests that both systems seem to perform well overall. Not only is the language quality acceptable, but the summarization systems seem to be able to *stay on-topic* (third statement) and *make entity-relevant summaries* (fourth statement), which might be the most crucial finding. However, Figure 6.3 and Figure 6.4 present an obvious difference: the bars for "Agree" are much taller for the abstractive system. The abstractive system has the majority of responses in "Agree" for every statement except the first. The extractive has the majority of responses split over "Somewhat agree" and "Agree", which is also very good. However, the abstractive system seems favorable based on the results in Figure 6.3 and Figure 6.4.

After introducing the following mapping: "Disagree" = 0, "Somewhat disagree" = 1, "Somewhat agree" = 2, and "Agree" = 3, the average value for the individual statements was calculated for the extractive and the abstractive system. Figure 6.5 presents the data in a diagram that can be used to compare the two summarization systems. The diagram further proves that the abstractive system outperforms the extractive system in every aspect. However, Figure 6.5 shows that the differences between the two systems are not that big, even though Figure 6.3 and Figure 6.4 might give the impression of the opposite. The largest difference between the systems appears in statements 1 and 3, suggesting that the abstractive system truly outshines the extractive system when it comes to language quality and ability to stay on-topic.

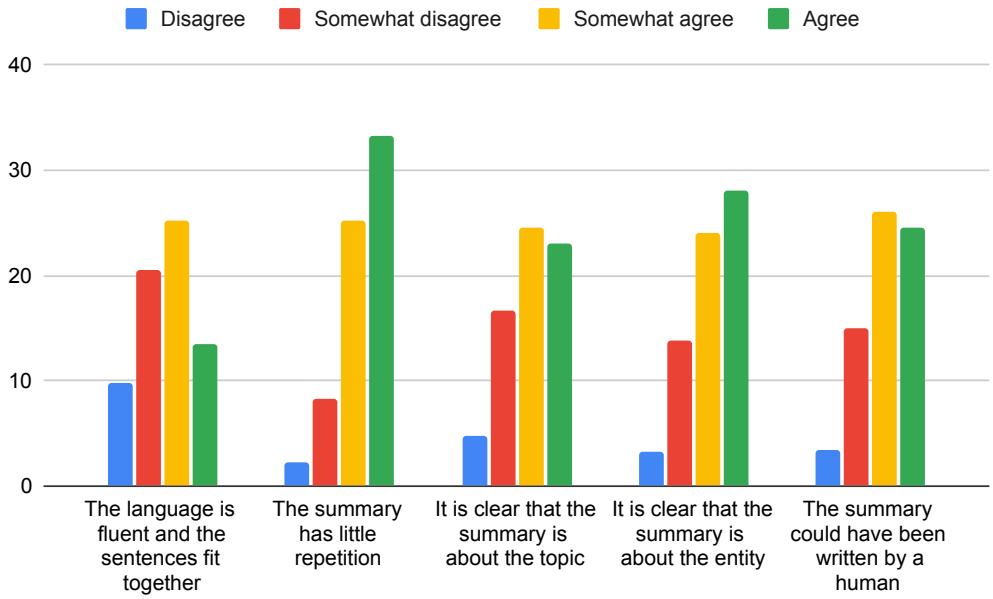



Figure 6.3.: Average response to the extractive summaries in the survey.

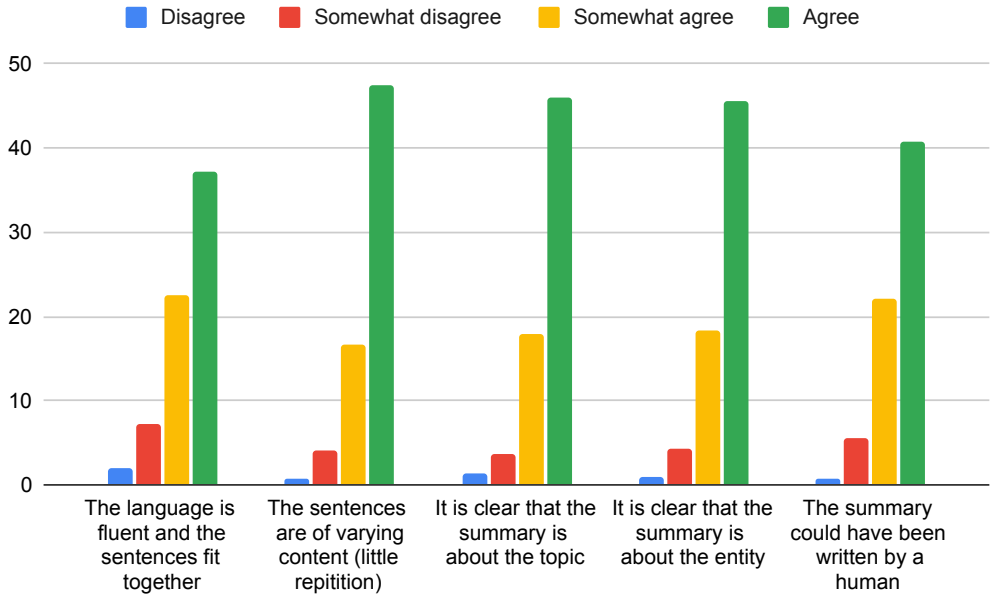



Figure 6.4.: Average response to the abstractive summaries in the survey.

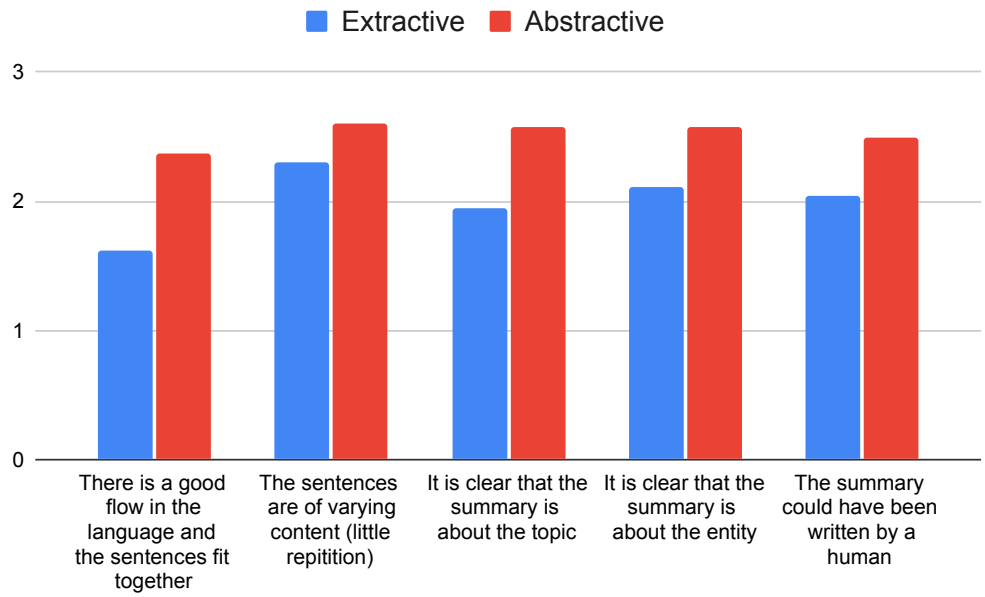



Figure 6.5.: Extractive and abstractive – average value for each statement using the mapping 0 = “Disagree”, 1 = “Somewhat disagree”, 2 = “Somewhat agree”, 3 = “Agree”.

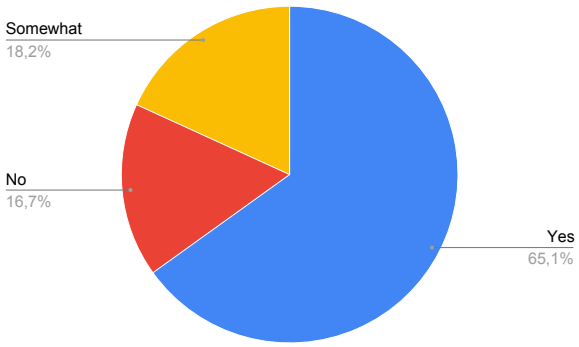



Figure 6.6.: Average distribution of answers to the question *Are you familiar with the topic?* (blue: "Yes", red: "No", yellow: "Somewhat").

#### *Are you familiar with the topic?*

Figure 6.6 presents the average distribution of responses to the question *Are you familiar with the topic?*. Only an average of 16.7% were unfamiliar with the topics. Thus, a total of 83.3% were familiar with the topics, although 18.2% out of the 83.3% were only somewhat familiar. This means that the goal of having topics that respondents were familiar with, as discussed in Section 6.3.1, is achieved. The respondents should thus have an idea of the content in the source documents, and consequently, the answers to statements three and four, as presented in Section 6.3.2, are more justified.

#### *Which summary do you prefer?*

As mentioned earlier in this section, the results regarding the statements (seen in 6.3 and Figure 6.4) suggests that the abstractive summaries are preferred. The answers to the question *Which summary do you prefer?* do not suggest otherwise. The average distribution of answers is presented in Figure 6.7. On average, 79.3% of respondents prefer the abstractive summary. What is interesting is that, on average, 17.5% prefers the extractive summary. The abstractive summaries are shorter and seemingly of better language quality, but a substantial number of respondents still prefer the extractive summaries. This suggests that there has to be something about the extractive summaries that makes some respondents prefer them.

#### *Could you imagine using a summarization system of this kind?*

The distribution of answers to the final question of the survey, *Could you imagine using a summarization system of this kind (for any purpose)?*, is presented in Figure 6.8. The results suggest that many can imagine themselves using an automatic summarization system. A total of 81.1% respondents say that they can see a use case for a summarization system like the ones proposed in this thesis, but note that 24.6% of these respondents answered "Yes, if it worked better". These findings suggest that, even though the majority

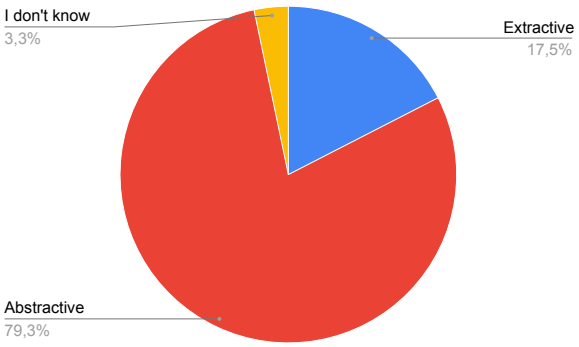



Figure 6.7.: Average distribution of answers to the question *Which summary do you prefer?* (red: "Abstractive", blue: "Extractive", yellow: "I don't know").

seems to like the summaries, the summarization systems have room for improvement, motivating future work within the field. It can also be noted that the 4.3% that answered "I don't know" most likely belongs to "No", as the question is very open-ended.

### The Best and The Worst Summary

The extractive system produced the poorest scoring summary, as presented in Figure 6.9. Characteristics of such a bad summary are mainly the lack of sentence coherence and the poor topic relevance. It can also be noted that the summary in Figure 6.9 consists of 88 words and is thus the longest summary in the survey.

The best scoring summary, presented in Figure 6.10, is generated by the abstractive system. The summary gets almost perfect scores for all statements, showing what the abstractive system is capable of. Characteristics of such a good summary are that it is concise, has good language quality, and has little repetition – all while staying topic-relevant and entity-related.

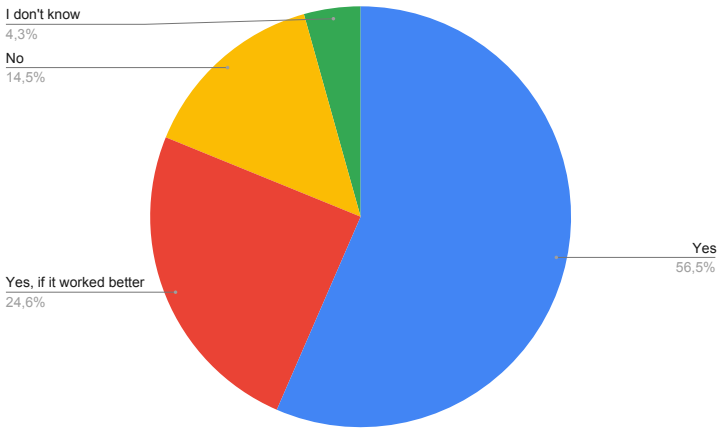



Figure 6.8.: Distribution of answers to the question *Could you imagine using a summarization system of this kind (for any purpose)?* (blue: "Yes", red: "Yes, if it worked better", yellow: "No", green: "I don't know").

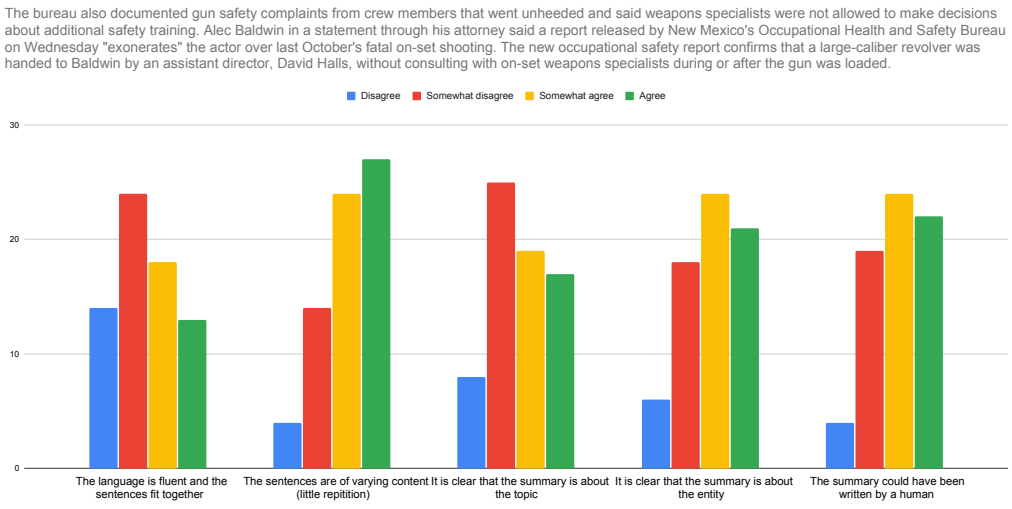



Figure 6.9.: The worst scoring summary (extractive).

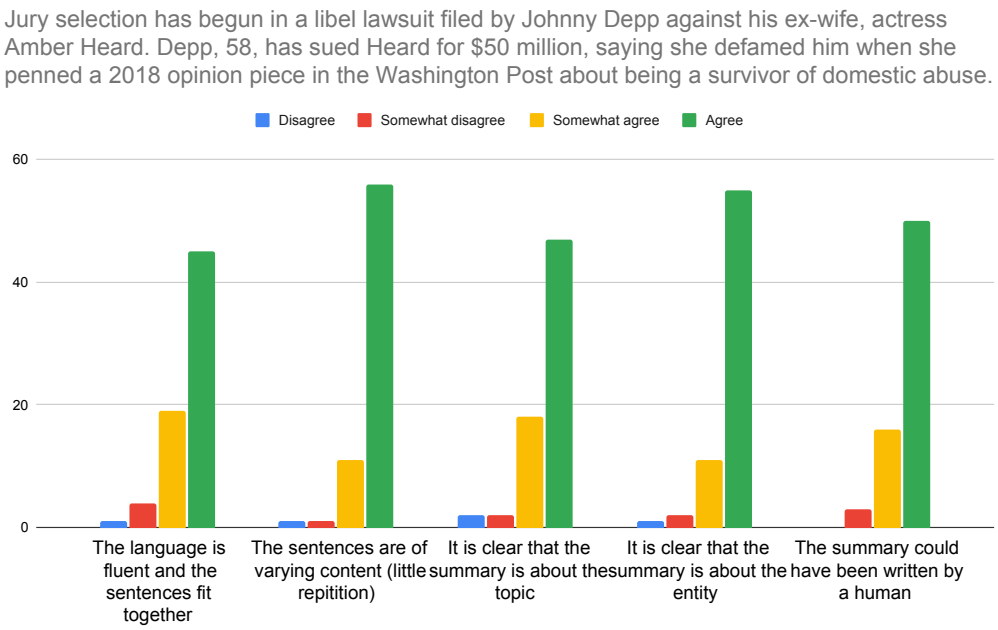



Figure 6.10.: The best scoring summary (abstractive).

# 7. Evaluation and Discussion

This chapter aims to thoroughly evaluate the summarization systems presented in Chapter 5 as well as discuss the research questions and the research goal presented in Section 1.2. Evaluation of the systems is primarily based on results from the conducted survey (Section 6.4) but also considers results from the automatic evaluation (Section 6.2). Further, the experiments' limitations and their implications on the evaluation are presented. The discussion will also shed light on other noteworthy aspects of the systems and consider the research questions and the goal in light of the evaluation. Note that some parts of this chapter use the same names for the summarization systems presented in Section 6.2.1.

## 7.1. Evaluating the Thesis Systems

This section intends to evaluate the systems, primarily based on the results from the survey. The extractive and abstractive system are evaluated Section 7.1.1 and Section 7.1.2 respectively.

### 7.1.1. Evaluating the Extractive System Based On Results

In the automatic evaluation (Section 6.2), Extractive (35) scores just below Abstractive MN. Neither of the two models is trained on CNN/DM; thus, no model should be at a disadvantage when comparing their results. The fact that Extractive (35) competes with Abstractive MN is promising for the extractive system, as Abstractive MN is arguably much more sophisticated. However, lead-3 significantly outperforms Extractive (35), which means that naïvely selecting the first three sentences is better than using the extractive model for the CNN/DM summarization task. But then again, Section 7.2.1 suggests that it is inadequate to use results from the automatic evaluation to draw conclusions.

The average response to the statements for extractive summaries, presented in Figure 6.5, suggests that the extractive system performs decently across the board. Recall that 0 = "Disagree", 1 = "Somewhat disagree", 2 = "Somewhat agree", 3 = "Agree". For the extractive system, the highest scoring statement in Figure 6.5 is the second, with a score of 2.30. This suggests that the goal of reducing repetition is achieved. However, it was hypothesized that the approach to reducing repetition would lead to poor sentence coherence, and that seems to be the case as the first statement in Figure 6.5 has the lowest score of 1.62. The scores for the rest of the statements lie at around 2, meaning that respondents "somewhat agree". Thus, the extractive system manages to perform its

task (entity-related multi-document summarization), but not to a particularly satisfying degree.

### 7.1.2. Evaluating the Abstractive System Based On Results

The average response to the statements for abstractive summaries, presented in Figure 6.5, suggests that the abstractive system succeeds at entity-related multi-document summarization. The scores for most statements are around 2.5, which is in the middle of "Somewhat agree" and "Agree". It is improbable that even a human could make summaries with a perfect score of 3.0 on the statements. In Zhao et al. (2020), human-made summaries reached an average score of 4.4 on a scale from 1 to 5 on similar statements, which translates to a score of 2.56 in the scale used in Figure 6.5. Thus, the results suggest that the abstractive system is very close to generating summaries of human quality.

It was speculated that the recursive approach would lead to repetitive summaries, as described in Section 5.4.4. The reason is that when different parts of the source documents are summarized, the summaries might contain the same information. Therefore, when the concatenation of those summaries is to be summarized, the resulting summary may be repetitive. However, no results from the survey indicate that this is the case, as the second statement in Figure 6.5 has the highest score out of all (2.61). Furthermore, as mentioned in Section 7.2.2, the abstractive summaries are generally short, which might be why there is so little repetition – it is unlikely to have redundant sentences if there are few sentences. However, the abstractive summary for the collection with $ID = 8$, which can be seen in Table A.2, contains four sentences and still has a decent score for the second statement (2.32).

It should be added that after the survey had been conducted, multiple respondents reached out and mentioned that they thought the survey was a kind of Turing test – where the goal was to identify which summary was made by a computer and which was written by a human. It is obvious that these respondents were referring to the abstractive summaries as being written by a human. This shows how sophisticated language generation technologies have become, and transformer models in particular, as human evaluators mistake machine-generated text as human-written.

## 7.2. Discussion

This section presents and discusses the experiments' limitations and noteworthy aspects of the systems that were not directly captured by the experiments. The limitations of the experiments say something about how justifiable the results from the experiments are and should be considered when revisiting the research questions. At the end of this section, the degree to which the research questions have been answered and the goal achieved is discussed.

### 7.2.1. Limitations of the Automatic Evaluation

Section 6.1 presents arguments for why the automatic evaluation approach was chosen, especially concerning the limitations of the CNN/Daily Mail dataset (CNN/DM) for evaluation. In short, the purpose of automatic evaluation was to gather insights regarding the systems' ability to compress textual information. Thus, CNN/DM was chosen as dataset and ROUGE as evaluation metric – because they are most commonly used to evaluate summarization systems (El-Kassas et al., 2021; Wang et al., 2017). However, the chosen dataset and evaluation metric presents serious limitations, which the following paragraphs present.

#### Limitations Regarding the Dataset

There are two main reasons why CNN/DM was a suboptimal choice as the evaluation dataset. The first reason is that CNN/DM is intended for single-document summarization (SDS), and thus, the multi-document summarization (MDS) systems from this thesis are expected to perform worse than state-of-the-art (SOTA) SDS systems. Section 6.1 argues that this does not really matter, as the goal is to compare the models' ability to compress textual information, multi-document or not. However, this brings up the second reason: Abstractive CNN/DM has an unfair advantage by being fine-tuned on CNN/DM. This was known beforehand, but the difference in performance was more significant than expected.

#### Limitations Regarding ROUGE Metrics

The ROUGE metrics are inherently flawed by design, as presented in 3.2.4. When measuring n-gram similarity (ROUGE-1 and ROUGE-2) or longest common subsequence similarity (ROUGE-L) between two summaries, the goal of a summarization model is to generate summaries that are as similar to the gold standard summaries as possible. However, two summaries can be vastly different regarding word choices but still convey the same information, as explained in Section 2.2. Thus, ROUGE fails to capture the inherent quality of a summary and can only be used as an approximation to measuring how good a summary truly is. It is worth mentioning that if there were a good way to measure the quality of a summary automatically, the summarization task would be much easier than it currently is, as one could build systems that optimize that metric.

#### What Do these Limitations Imply?

The ROUGE scores for the thesis systems are significantly lower than those of SOTA models, as seen in Table 6.2. Abstractive CNN/DM is the exception, which is close to the SOTA models, but still outperformed by the regular PEGASUS CNN/DM, which is the same model but without chunking. However, the limiting factors imply that comparing the systems to SOTA models for CNN/DM is futile because (1) the systems are not designed for SDS and (2) the ROUGE metric is flawed. Nevertheless, the results can be

used to compare Abstractive MN and the extractive system, as neither is fine-tuned on CNN/DM.

In retrospect, it can be argued that the automatic evaluation's limitations outweigh the results' significance, which renders the experiment somewhat unnecessary. However, it must be emphasized that the purpose of the automatic evaluation was to have a low-effort way of comparing the systems' ability to summarize text. In the end, the most important results will come from the survey.

### 7.2.2. Limitations of the Survey

The survey gathered a lot of data that provides a sound basis for evaluating the summarization systems. However, some significant limitations should be acknowledged before the evaluation is performed, and they are presented in the following paragraphs.

#### Limitations Concerning Demographics

When it comes to data quantity, more is always better. The survey had a total of 69 respondents, which is not a massive number of people, but enough to observe statistical trends and indications regarding the summarization systems. Furthermore, as mentioned in Section 6.4.1, many respondents are classmates and employees at Strise. Consequently, the average respondent is likely to be more interested in technology than the average person, which means that answers to the question "*Could you imagine using a summarization system of this kind (for any purpose)?*" are likely skewed towards positive answers ("Yes" and "Yes, if it worked better").

Respondents of the survey can be classified as *WEIRD* – coming from a western, educated, industrialized, rich, and democratic society. Reaching out to a WEIRD audience was inevitable, given the scope of this thesis. Since WEIRD societies only make up 12% of the world population (Azar, 2010), the diversity of the survey results are somewhat limited. Consequently, some answer distributions are likely skewed. For instance, this demographic is probably more likely to answer "Yes" to the question "*Could you imagine using a summarization system of this kind (for any purpose)*". However, it should be noted that there is a diverse gender and age distribution among the respondents, as presented in Figure 6.1.

#### Limitations Concerning Respondents Familiarity With the Source Documents

A major limitation of the survey is the fact that it is infeasible to ask respondents to read the source documents of summaries; simply because there are too many of them. To combat this issue, the topics of the news article collections were specifically chosen to be something the respondents might be familiar with. Thus, respondents expect what a summary should contain despite not reading the source documents. This is described in more detail in Section 6.3.1. However, it was expected that not all respondents were familiar with every topic, and Figure 6.6 shows that only 65.1% of respondents were familiar with the topics on average. Consequently, the statements regarding the

summaries presented in Section 6.3.2 were made such that even someone unfamiliar with the topic should be able to answer them. This shows a shortcoming of the survey, as it did not utilize respondents familiar with the topics. For instance, follow-up questions like "*If you are familiar with the topic, would you say this summary is a good summary of the topic?*" could have been asked.

The fact that respondents have not read the source documents also makes it hard to determine whether a summary is a good entity-related summary. The fourth statement ("*It is clear that the summary is about the entity*") is intended to check if the summary focuses on the entity. However, a summary that mentions the specified entity often is likely to score high on the fourth statement but is not necessarily a good entity-related summary. Thus, it is just an assumption that a summary is entity-related if it is both on-topic (third statement – *"It is clear that the summary is about the topic"*) and concerns the entity (fourth statement).

### Limitations Concerning the Working Dataset and Choosing Collections

The working dataset, as presented in Section 4.1, represents the real-world data the systems are intended for. News articles in the dataset are scraped from online websites and thus may contain a lot of junk. Some actions are taken to remove junk in the data preparation phase, as explained in Section 5.2. However, some junk may remain after data preparation; one example is non-English sentences. In fact, when summarizing the collection with $id = 10$ in the working dataset and taking Mark Zuckerberg as input entity, the extractive system extracts one Spanish sentence. This summary can be seen in Table A.2 and would have been included in the survey if collection with $id = 10$ had been chosen.

If many summaries containing junk were included in the survey, the results would likely be much worse, especially for the extractive summaries that are most prone to including junk. It can thus be argued that the summary examples in the survey were cherry-picked. However, a counterargument is that the summaries in the survey were chosen based on the topic of the collections and not on summary quality, as explained in Section 6.3.1. The first summary in the survey even contains a part of an ad, which is the extractive summary of the collection with $id = 0$ and can be seen in Table A.2.

In essence, this paragraph implies that the varying quality of source documents, especially in terms of containing junk, may negatively impact the quality of generated summaries. However, the summaries in the survey mostly seem to be devoid of junk, which either suggests that the data preparation is effective or that the selection of summaries in the survey is too small. Nevertheless, it is worth noting that examples like the extractive summary for the collection with $id = 10$ may occur, making the extractive system somewhat unreliable.

### Limitations Concerning the Summaries in the Survey

There are some minor but noteworthy limitations regarding the generated summaries presented in the survey. These limitations include:

**Few summary instances** A test-respondent suggested that four summary topics were optimal for the survey. With regard to number of respondents, the survey focused on quantity rather than quality; thus, four summaries were regarded as satisfactory. However, it must be noted that the four example summaries per system do not necessarily represent the actual average performance of the systems. Consequently, the results have somewhat limited coverage. This ties together with the limitations concerning the working dataset, as discussed in the previous paragraph.

**Only single-entity summaries** Every summary presented in the survey is based on one entity. Thus, the survey provides no ground for discussing entity-summarization with multiple entities. However, entity-summarization with one input entity is considered the main use case of the systems.

**No human-written summaries** To keep the survey short, no human-written summaries were included. However, it would be valuable to see how respondents would evaluate summaries written by a human. The abstractive system scores very well, and it would be interesting to compare the results to a human baseline.

**Unequal summary lengths** When using the default parameters of the systems, the extractive system generally produces longer summaries than the abstractive system. In the survey, the extractive summaries have an average length of 78 words, while the abstractive summaries have an average of 53 words. It can be assumed that humans prefer shorter summaries because they are faster to read; thus, the length difference might impact the results in favor of the abstractive system. In particular, results from the question about which summary respondents prefer may be skewed in favor of the abstractive system because of the summary lengths.

#### What Do these Limitations Imply?

The primary takeaway from this section is that results from the survey are somewhat limited with regard to which conclusions can be drawn from them. For instance, the results cannot conclude whether a system is truly good at generating entity-related summaries. However, results still provide a solid basis for evaluation and discussion, as long as the limitations are considered.

### 7.2.3. Other Aspects of the Systems

The following paragraphs present topics and major limitations regarding the summarization systems that have hitherto not been discussed.

#### Limiting Aspects of the Data Preparation Phase

The major limitations regarding data preparation are closely connected to the limitations of the working dataset, as presented in Section 4.1.2. First, sentences that mention the input entity, and are thus summary-relevant, might be filtered out unintentionally. This can be due to multiple reasons:

1. The Named Entity Recognition (NER) system may not identify all entity mentions.
2. The NER system can misidentify entities, like the case discussed in Section 4.1.2 where Chris Rock was identified as rock music
3. Entity mentions that are not named, e.g., "he" and "she", will not be recognized by the NER system. Thus, coreference resolution, described in Section 2.2, might be fitting for future work.

A sentence tokenizer is used to split sentences during data preparation. However, the tokenizer is not perfect, and it struggles with some of the scraped data in the working dataset. An example of this is shown in the first sentence of the first summary in the survey: "*FAIRFAX, Va. (AP) — A trial over libel allegations by Johnny Depp against his ex-wife*, [...]". The tokenizer recognized "*FAIRFAX, Va. (AP)*" as a part of the sentence when it is actually a part of an ad prior to the sentence. Thus, it shows how the tokenizer might wrongly divide sentences.

### Limiting Aspects of the Extractive Approach

The extractive design (Section 5.3) is based on some noteworthy assumptions. The first assumption is that the sentence encoder, SentenceBERT, captures the semantic meaning of sentences. The second assumption is that K-means clustering groups semantically similar sentences, which is especially arduous when considering the difficulty of finding the optimal K for the K-means algorithm (El-Kassas et al., 2021). Lastly, it is assumed that TextRank will identify the most representative sentence in clusters from K-means. The assumptions made, presented i more detail in Section 5.3, are logically sound. However, it should be noted that these assumptions are not always true, and erroneous summaries can be produced as a result.

### Limiting Aspects of the Abstractive Approach

The chunking algorithm (Section 5.4.2) is flawed because small chunks receive the same weighting as large chunks. For instance, the if there are 600 tokens in a recursive step, the first chunk might get 500 tokens and the second chunk gets 100. However, both of these chunks will be summarized to 50-80 tokens (by default) and concatenated, which means that both chunks are regarded as equally important, despite the first chunk having five times as many tokens.

The abstractive system could also have used larger chunks. It was initially believed that the PEGASUS models provided by HuggingFace were based on the base version, $\text{PEGASUS}_{\text{BASE}}$ (Zhang et al., 2019). However, after the experiments it was discovered that the PEGASUS models fine-tuned on CNN/DM and MN respectively are based on $\text{PEGASUS}_{\text{LARGE}}$ (Zhang et al., 2019), which has a maximum input size of 1024 tokens. Thus, the chunking algorithm in the abstractive system could have made chunks of maximum 1024 tokens instead of 512 as it currently does. Doing so would likely increase the ROUGE score for the CNN/DM experiment, as it would be closer to what Zhang

et al. (2019) did. However, larger chunks do not necessarily make a big difference when the number of chunks grows, as the summaries from the chunks would be recursively summarized anyway. Thus, the impact of increasing the chunk size is deemed minimal with regard to the survey. It is doubtful that larger chunks size would increase the performance by a substantial amount, considering how good the results are.

It is worth mentioning that the literature review in Section 3.2.3 identified papers that mention flaws of recursive approaches. For instance, Wu et al. (2021) report that errors in the first level of recursion tend to propagate throughout the recursive steps, meaning that errors at early stages often end up in the final summary.

#### Time Consumption

Table A.2 presents summaries generated by the systems, as well as the associated summarization time. The extractive system typically uses 10 to 20 seconds depending on the input size, while the abstractive system typically uses 2 to 4 minutes. Thus, although the abstractive system is preferred in around 80% of cases according to human evaluators (Figure 6.7), Table A.2 shows that it is usually more than ten times slower than the extractive system. Note that the summaries were generated on a Macbook Pro 2019, and better hardware would speed up the process.

### 7.2.4. Revisiting the Research Questions and the Goal

The research questions and the goal for this thesis, presented in Section 1.2, laid the foundation for the thesis project. Thus, in the aftermath of experimentation and evaluation, the questions and the goal are revisited to see whether they have been answered or achieved to a satisfying degree.

**Research question 1 (RQ1)** *Does removing sentences that do not include a specific entity and using a state-of-the-art summarization model to summarize the remaining text produce a good entity-related summary?*

The answers to the statements in the survey, which can be seen in Table 6.5, are highly relevant for RQ1. In particular, the third and fourth statements can be used in combination to determine if a summary is entity-related. The fourth statement, "*it is clear that the summary is about the entity*", is clearly intended to check if the summary text relates to an entity. However, if the summary is entity-related but does not stay on-topic, it is not a good summary. Thus, the fourth statement must be combined with the third statement; "*it is clear that the summary is about the topic*". The abstractive system scores exceptionally well on both statements, and of course, it incorporates the naïve approach of removing irrelevant sentences. Thus, the survey suggests that the approach works for creating entity-related summaries from a collection of news articles, which means that the answer to RQ1 seems to be "yes".

It must be noted that the extractive system does not score as well as the abstractive system on the third and fourth statements. However, this is most likely because the

abstractive system is generally better, which is supported by pretty much all results presented in this thesis. Thus, it is not necessarily an indication that the naïve approach does not work, and the scores are still *positive* (i.e., closer to "Agree" than "Disagree"). At the same time, it must be mentioned that the survey has some limitations that must be considered when interpreting the results, as discussed in Section 7.2.2. The fact that respondents have not read the source documents means they cannot truly judge whether a summary is a good entity-related summary. Thus, results from the survey only suggest that the approach works, but it cannot be stated as a factual statement that the approach provides good entity-related summarization.

**Research question 2 (RQ2)** *Can the summarization models be considered state-of-the-art for automatic summarization tasks?*

Recall that "summarization model" refers to the part of the systems after data preparation. PEGASUS (Zhang et al., 2019) achieves SOTA results on multiple datasets, including CNN/DM and Multi-News. Further, the abstractive design in this thesis does not modify the PEGASUS model – it only applies it in a recursive approach. Thus, it is reasonably safe to conclude that the abstractive model can be considered state-of-the-art for standard summarization tasks. On the other hand, the extractive model is a fusion of multiple elements hypothesized to create non-redundant summaries (Section 5.3). Although the model seems to be able to reduce redundancy, it does not score exceptionally well on the CNN/DM experiment (Section 6.2) and does not receive outstanding results from the survey (Section 6.4). Despite considering the experiments' limitations, the results indicate that it is not on par with SOTA summarization models. Thus, the extractive system is not considered state-of-the-art for standard summarization tasks. However, it performs decently and has other advantages, like being much faster than the abstractive model.

**Research question 3 (RQ3)** *Are the summarization systems suitable for a real-world scenario?*

"Real-world scenarios" refers to cases with more aspects to account for than just summary quality, in contrast to in-development, where the goal may be to maximize ROUGE scores for a dataset. An example of a real-world scenario is to generate an entity-related summary on the fly, given a collection of news articles. Especially important in a real-world scenario is thus low time consumption and consistent performance.

First of all, it should be noted that the data preparation phase is practically instantaneous. However, this is only when the entity recognition work is done beforehand, which is the case for Strise. Thus, it is assumed that no extra time is used for entity recognition, which would otherwise have to be accounted for.

The time consumption of the summarization models is discussed in Section 7.2.3, which shows that the systems are generally slow – especially the abstractive model. The PEGASUS transformer used in the abstractive model is large and computationally heavy, making the current abstractive design unfit for real-world scenarios. At the same time, it

must be noted that the PEGASUS transformer is the sole reason why the abstractive system scores so well in every regard on the survey. Also, results from experiments suggest that the abstractive system consistently generates good summaries. However, regarding RQ3, a conclusion for the abstractive system is that the current design is way too slow to be used in a real-world scenario, despite the quality of generated summaries. Some system modifications can reduce the computation time, like shrinking the number of source documents or using a smaller transformer model. However, such modifications will likely reduce the summary quality. Thus, inspecting ways of making the abstractive model faster is a suitable topic for future work.

The extractive model is much faster than the abstractive. However, it typically uses around 10-20 seconds to summarize a collection in the working dataset, as seen in Table A.2. Thus, despite being the faster model, the extractive model is still a bit slow for a real-world scenario. Further, the model seems to be somewhat inconsistent, as it has varying results in the survey and large standard deviations for the ROUGE scores in the automatic evaluation. In the same way as the abstractive model, some modifications can make it quicker, like using a smaller model for sentence embedding.

To answer RQ3: the systems designed and implemented in this thesis seem too slow for a real-world scenario where time is an essential aspect. However, note that summarization times were measured on a Macbook Pro 2019. Of course, better hardware will improve the efficiency, but the current time measurements are deemed a fair baseline. One thing to add is that this section only considers cases where an input entity is given. If none is given, the number of candidate sentences rises significantly, making the models even slower and less suitable for a real-world scenario.

Finally, it is worth mentioning that certain modifications to the systems can make summarization faster, although they might reduce summary quality. Consequently, improving efficiency is a fitting topic for future work.

**Research question 4 (RQ4)** *Are the summaries produced by the summarization systems satisfactory according to human evaluators?*

RQ4 is in large parts answered by the evaluations of the summarization systems in Section 7.1.1 and Section 7.1.2. To recap: the extractive system seems good at avoiding redundancy and decent at staying on-topic. However, it seems to struggle with the language flow and making the sentences fit together. Overall, human evaluators say they "somewhat agree" that the summaries produced by the extractive system are of human quality. On the other hand, the abstractive system is seemingly excellent concerning every aspect mentioned above. Results for the abstractive summaries are almost on par with results from human-made summaries presented in Zhao et al. (2020).

To answer RQ4: the abstractive summaries are clearly satisfactory according to human evaluators. However, it is harder to conclude whether extractive summaries are satisfactory, as they are poor in some aspects but are acceptable in others. Thus, the extractive summaries are considered somewhat satisfactory.

**Goal** *Provide a novel approach to automatic entity-related summarization based on a collection of text documents.*

As no prior work for entity-related summarization from text was identified during the literature search, a naïve solution has been proposed: given a collection of documents and an entity, filter out every sentence that does not mention the entity and perform summarization in a typical manner on the remaining text. To verify that this approach works, two different summarization systems using the approach have been implemented (one extractive and one abstractive), and experiments are conducted to create a basis for evaluation. According to results, primarily from the survey that was conducted, the systems seem to perform the task of entity-related summarization well. Thus, one may conclude that the goal has been reached.

# 8. Conclusion and Future Work

The final chapter intends to wrap up the work presented in the previous chapters, and thus, Section 8.1 summarizes what has been done. Further, Section 8.2 discusses the contributions made with regard to the research questions, and Section 8.3 proposes ideas for future work identified during this project.

## 8.1. Conclusion

The goal of this thesis was to propose a novel approach to entity-related multi-document summarization. No prior work for the specific purpose of generating summaries concerning entities based on a collection of text was identified during the literature search, and thus, a naïve approach has been proposed. The approach is to, given a collection of documents and an entity, (1) use Named Entity Recognition (NER) to identify entity mentions, (2) remove sentences that do not mention the given entity, and (3) summarize the remaining sentences with a multi-document summarization system. The thesis is thus about designing and implementing such systems and then evaluating them to see if the approach works.

The NER system used in this thesis is provided by Strise[1]. Thus, a dataset containing collections of news articles and entity-mentions in the articles was created to represent the desired use case of the summarization systems – to create entity-related summaries from multiple news articles.

Two types of summarization systems are proposed: one extractive and one abstractive. The extractive system uses a clustering approach primarily inspired by Zhao et al. (2020), focusing on minimizing repetition. The abstractive system utilizes a state-of-the-art (SOTA) transformer model (PEGASUS) with a recursive approach inspired by Sinha et al. (2018) to compensate for the transformer's limited input size.

Both automatic evaluation and a survey have been conducted to create a basis for evaluating the systems. The automatic evaluation consisted of calculating ROUGE-1, ROUGE-2, and ROUGE-L scores for the CNN/Daily Mail test set. Due to the chosen dataset and ROUGE's inherent flaws, the results are seriously limited. However, they suggest that the extractive system can compete with the abstractive system when it comes to compressing textual information. On the other hand, the survey provided interesting results regarding different aspects of the systems: language quality, redundancy in summary, and ability to stay on-topic and entity-related. Results clearly show that respondents prefer the abstractive system. The abstractive system gets close to the top

[1]The company that the thesis project was done in collaboration with – https://www.strise.ai/

score for every aspect. The extractive system also gets positive scores, but substantially less than the abstractive system, and it seems to struggle with sentence coherence.

The limiting factors of the conducted experiments make it hard to conclude whether the systems truly create good entity-related summaries. Nevertheless, the results from the experiments suggest that the systems work for their intended use case, and the majority of survey respondents reported that they were interested in using such a summarization system (for any purpose). Thus, the goal seems to have been reached.

## 8.2. Contributions

This section describes the main contributions of the conducted work:

**An approach to entity-related summarization** In short, the approach to generating entity-related summaries consists of removing sentences that do not mention the specified entity and summarizing the remaining sentences in a typical manner. Both summarization systems in this thesis implement this approach. Research question 1 asks whether this approach works, and according to survey results, it seems that the systems are able to generate entity-related summaries (especially the abstractive system).

**A model for extractive multi-document summarization** Section 5.3 presents the extractive model, which uses a SOTA sentence encoder and a clustering approach to extract important sentences from the source documents. The implementation minimizes repetition among the sentences but seems to result in poor sentence coherence. Human evaluators found the extractive model decent overall but still well away from producing human-quality summaries.

**A model for abstractive multi-document summarization** Section 5.4 presents the abstractive model, which uses a SOTA transformer model to summarize the source documents recursively. The recursive approach is implemented to overcome the input size limit of the transformer model. The abstractive system achieves excellent scores in the survey, and evaluation suggests that it generates summaries close to human quality.

**A survey design for entity-related summarization** Section 6.3 presents a possible survey design that lays the foundation for collecting data to evaluate entity-related summarization systems. An example implementation of the survey design can be seen in Appendix C. A thorough discussion concerning shortcomings of the survey and limitations of the results are provided in Section 7.2.2.

**Results from a survey** A significant contribution is empirical data that have been collected by conducting a survey utilizing the aforementioned survey design. This data is well-suited for evaluating the summarization systems.

**An evaluation of the systems in light of the survey results** Section 7.1 and Section 7.2.4 provides a thorough evaluation of the two summarization systems based on results from the survey. The most significant finding is that the abstractive system seems to generate human-quality summaries according to human evaluators.

## 8.3. Future Work

The work in this thesis has presented ideas for future work within the field of entity-related summarization. The ideas mainly consist of improving the systems presented in this thesis or trying out entirely new approaches to entity-related summarization. This section presents each idea for future work.

### 8.3.1. Make Multilingual Systems

When the working dataset presented in Section 4.1 were created, a lot of non-English articles were filtered out. From a business standpoint, it could be valuable to make an entity-related summarization system that handles non-English languages in addition to English. It should not be too difficult to make the systems presented in this thesis work on collections of articles in a non-English language. Firstly, the data preparation (Section 5.2) must differ depending on the language. For the extractive system (Section 5.3), the only language-dependent part is the BERT sentence encoder. Thus, by exchanging the encoder with a multilingual encoder (e.g., Facebook Research's LASER (Artetxe and Schwenk, 2019)), the extractive system should, in theory, work for non-English languages. The same goes for the abstractive system (Section 5.4); implementing a multilingual transformer model should enable the system to generate non-English summaries. Note that the premise for this idea is that every document in a collection is in the same language. The opposite case of having a collection of articles in multiple languages might prove more difficult but is still an interesting topic.

### 8.3.2. Use Coreference Resolution to Identify more Candidate Sentences

As mentioned in Section 7.2.3, coreference resolution can be used to identify mentions of an entity that a Named Entity Recognition (NER) system misses. The data preparation step (Section 5.2) uses a NER system to find mentions of entities in the source documents. However, mentions like "him" or "her" will not be included. Thus, many potentially summary-relevant sentences are lost, and coreference resolution can be used to address this problem.

### 8.3.3. Conduct More In-Depth Human Evaluation

The survey conducted in this thesis provides a limited basis for drawing conclusions regarding the performance of the entity-related summarization systems. In future work, more in-depth human evaluation should be conducted. Preferably, interviews should be

held, where each test subject is asked to read the source documents and then rate the generated summaries. Of course, some measures should be made to make the interviews more feasible, like reducing the number of source documents in a collection. A human evaluation of this kind would give more detailed insights into the summarization systems' ability to produce entity-related summaries.

### 8.3.4. Explore Knowledge Graph Approaches

As mentioned in Section 3.1, "entity summarization" refers to generating a summary of an entity based on linked data in a knowledge graph. It is not uncommon for text summarization approaches to utilize a graph structure to represent the textual data and then apply a graph summarization algorithm. The same approach could be taken for entity-related summarization: generate a knowledge graph from the source documents and then apply a SOTA entity summarization method.

### 8.3.5. Explore Question-Answering System Approaches

Much research has been done in the field of automatic question answering, and one may argue that the field is very similar to entity-related summarization. For instance, one could define entity-related summarization as answering the question "what happened to *entity*?" Thus, a SOTA question-answering system could be used to generate an answer to that question, which would then be considered as the *summary*. Since question answering is a well-researched field, it would be interesting to see how well the task could be translated to entity-related summarization.

### 8.3.6. Explore Ways Of Making the Abstractive System More Efficient

As discussed in Section 7.2.4 (research question 3), the abstractive system is very slow, which for a real-world use case is undesirable. However, abstractive summaries seem preferable among human evaluators (Section 6.4), and many actions could be taken to speed up the summarization process. For instance, the number of input documents could be reduced by identifying the n most important documents, or a smaller and faster transformer model could be implemented.

### 8.3.7. Explore the New EntSUM Dataset

EntSUM is a dataset published in April 2022 (Maddela et al., 2022) but was discovered too late to impact this thesis. EntSUM is a human-annotated dataset based on The New York Times Annotated Corpus dataset (NYT)[2]. Each data instance primarily consists of (1) the ID of the source document in NYT, (2) the relevant entity, and (3) a summary relating to the entity. Thus, it appears that this dataset can be used for entity-related summarization (e.g., evaluation), and future work should consider this dataset. Note that this dataset is for single-document summarization.

[2] https://catalog.ldc.upenn.edu/LDC2008T19

# Bibliography


Samer Abdulateef, Naseer Ahmed Khan, Bolin Chen, and Xuequn Shang. Multidocument arabic text summarization based on clustering and word2vec to reduce redundancy. *Information*, 11(2), 2020. ISSN 2078-2489. doi:10.3390/info11020059. URL https://www.mdpi.com/2078-2489/11/2/59.

Mikel Artetxe and Holger Schwenk. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, sep 2019. ISSN 2307-387X. doi:10.1162/tacl_a_00288. URL https://doi.org/10.1162/tacl_a_00288.

Kevin Arvai, Aldren Santos, David Amos, Geir Arne Hjelle, Joanna Jablonski, and Jacob Schmitt. K-means clustering in python: A practical guide, 2020. URL https://realpython.com/k-means-clustering-python/.

B. Azar. Are your findings 'weird'?, May 2010. URL https://www.apa.org/monitor/2010/05/weird.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL https://arxiv.org/abs/1409.0473.

Shikha Bordia and Samuel R. Bowman. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi:10.18653/v1/N19-3002. URL https://aclanthology.org/N19-3002.

Léo Bouscarrat, Antoine Bonnefoy, Thomas Peel, and Cécile Pereira. Strass: A light and effective method for extractive summarization based on sentence embeddings, 2019. URL https://arxiv.org/abs/1907.07323.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Hiram Calvo, Pabel Carrillo-Mendoza, and Alexander Gelbukh. On redundancy in multi-document summarization1. *Journal of Intelligent & Fuzzy Systems*, 34:1–11, may 2018. doi:10.3233/JIFS-169507.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. Faithful to the original: Fact aware neural abstractive summarization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), apr 2018. doi:10.1609/aaai.v32i1.11912. URL https://ojs.aaai.org/index.php/AAAI/article/view/11912.

Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany, August 2016. Association for Computational Linguistics. doi:10.18653/v1/P16-1223. URL https://aclanthology.org/P16-1223.

Bobby Chernev. How many blogs are published per day in 2022?, June 2022. URL https://techjury.net/blog/blogs-published-per-day.

Wafaa S. El-Kassas, Cherif R. Salama, Ahmed A. Rafea, and Hoda K. Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2020.113679. URL https://www.sciencedirect.com/science/article/pii/S0957417420305030.

Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model. *CoRR*, abs/1906.01749, 2019. URL http://arxiv.org/abs/1906.01749.

Alexios Gidiotis and Grigorios Tsoumakas. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:3029–3040, 2020. doi:10.1109/TASLP.2020.3037401.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2019. URL https://arxiv.org/abs/1904.09751.

Oleksandra Klymenko, Daniel Braun, and Florian Matthes. Automatic text summarization: A state-of-the-art review. pages 648–655, jan 2020. doi:10.5220/0009723306480655. URL https://www.researchgate.net/publication/341469411_Automatic_Text_Summarization_A_State-of-the-Art_Review.

Anders Kofod-Petersen. How to do a structured literature review in computer science. Technical report, Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway, 2015.

Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset. *CoRR*, abs/1810.09305, 2018. URL http://arxiv.org/abs/1810.09305.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 540–551, Hong Kong, China, November 2019. Association for Computational Linguistics. doi:10.18653/v1/D19-1051. URL https://aclanthology.org/D19-1051.

Sayali Kulkarni, Sheide Chammas, Wan Zhu, Fei Sha, and Eugene Ie. Aquamuse: Automatically generating datasets for query-based multi-document summarization, 2020. URL https://arxiv.org/abs/2010.12694.

Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/677e09724f0e2df9b6c000b75b5da10d-Paper.pdf.

Logan Lebanoff, Kaiqiang Song, and Fei Liu. Adapting the neural encoder-decoder framework from single to multi-document summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4131–4141, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi:10.18653/v1/D18-1446. URL https://aclanthology.org/D18-1446.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. URL https://arxiv.org/abs/1910.13461.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. *CoRR*, abs/2011.05864, 2020. URL https://arxiv.org/abs/2011.05864.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Qingxia Liu, Gong Cheng, and Yuzhong Qu. Deeplens: Deep learning for entity summarization, 2020. URL https://arxiv.org/abs/2003.03736.

Qingxia Liu, Gong Cheng, Kalpa Gunaratna, and Yuzhong Qu. Entity summarization: State of the art and future challenges. *Journal of Web Semantics*, 69:100647, 2021. ISSN 1570-8268. doi:https://doi.org/10.1016/j.websem.2021.100647. URL https://www.sciencedirect.com/science/article/pii/S1570826821000226.

Yixin Liu and Pengfei Liu. Simcls: A simple framework for contrastive learning of abstractive summarization. *CoRR*, abs/2106.01890, 2021. URL https://arxiv.org/abs/2106.01890.

Niyaz M. Salih and Karwan Jacksi. State of the art document clustering algorithms based on semantic similarity. *Jurnal Informatika*, 14:58–75, may 2020. doi:10.26555/jifo.v14i2.a17513.

Mounica Maddela, Mayank Kulkarni, and Daniel Preotiuc-Pietro. EntSUM: A data set for entity-centric extractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3355–3366, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-long.237.

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-3252.

Ramesh Nallapati, Bing Xiang, and Bowen Zhou. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023, 2016. URL http://arxiv.org/abs/1602.06023.

N. Nazari and M. A. Mahdavi. A survey on automatic text summarization. *Journal of AI and Data Mining*, 7(1):121–135, 2019. ISSN 2322-5211. doi:10.22044/jadm.2018.6139.1726. URL http://jad.shahroodut.ac.ir/article_1189.html.

Ani Nenkova. Summarization evaluation for text and speech: issues and approaches. In *Ninth International Conference on Spoken Language Processing*, Pittsburgh, Pennsylvania, sep 2006.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9, Montréal, Canada, June 2012. Association for Computational Linguistics. URL https://aclanthology.org/W12-2601.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018. doi:10.18653/v1/n18-1049. URL https://doi.org/10.18653%2Fv1%2Fn18-1049.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *CoRR*, abs/1705.04304, 2017. URL http://arxiv.org/abs/1705.04304.

Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter*

*of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi:10.3115/v1/N15-1026. URL https://aclanthology.org/N15-1026.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019. URL http://arxiv.org/abs/1908.10084.

Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010. ISBN 978-0-387-30164-8. doi:10.1007/978-0-387-30164-8_832. URL https://doi.org/10.1007/978-0-387-30164-8_832.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks, 2017.

Aakash Sinha, Abhishek Yadav, and Akshay Gahlot. Extractive text summarization using neural networks. *CoRR*, abs/1802.10137, 2018. URL http://arxiv.org/abs/1802.10137.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2020. URL https://arxiv.org/abs/2009.01325.

Paulus Setiawan Suryadjaja and Rila Mandala. Improving the performance of the extractive text summarization by a novel topic modeling and sentence embedding technique using sbert. In *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–6, 2021. doi:10.1109/ICAICTA53211.2021.9640295.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. URL http://arxiv.org/abs/1409.3215.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 305–312, 2017. doi:10.1109/BigDataCongress.2017.46.

Dongjun Wei, Yaxin Liu, Fuqing Zhu, Liangjun Zang, Wei Zhou, Jizhong Han, and Songlin Hu. Esa: Entity summarization with attention, 2019. URL https://arxiv.org/abs/1905.10625.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback, 2021. URL https://arxiv.org/abs/2109.10862.

Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), apr 2018. doi:10.1609/aaai.v32i1.11987. URL https://ojs.aaai.org/index.php/AAAI/article/view/11987.

Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *CoRR*, abs/1601.01343, 2016. URL http://arxiv.org/abs/1601.01343.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. 2020. doi:10.48550/ARXIV.2007.14062. URL https://arxiv.org/abs/2007.14062.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777, 2019. URL http://arxiv.org/abs/1912.08777.

Jinming Zhao, Ming Liu, Longxiang Gao, Yuan Jin, Lan Du, He Zhao, He Zhang, and Gholamreza Haffari. Summpip: Unsupervised multi-document summarization with sentence graph compression. *CoRR*, abs/2007.08954, 2020. URL https://arxiv.org/abs/2007.08954.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. Extractive summarization as text matching. *CoRR*, abs/2004.08795, 2020. URL https://arxiv.org/abs/2004.08795.

Hao Zhou, Weidong Ren, Gongshen Liu, Bo Su, and Wei Lu. Entity-aware abstractive multi-document summarization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 351–362, Online, August 2021. Association for Computational Linguistics. doi:10.18653/v1/2021.findings-acl.30. URL https://aclanthology.org/2021.findings-acl.30.

| ID | Collection topic | Relevant entity |
|---|---|---|
| 0 | Johnny Depp and Amber Heard defamation trial | Johnny Depp |
| 1 | Elon Musk is planning to buy Twitter | Elon Musk |
| 5 | Will Smith slapping Chris Rock on-stage during the Oscars 2022 | Will Smith |
| 7 | Rust shooting incident: a fatal on-set shooting where Alec Baldwin shot cinematographer Halyna Hutchins with a presumably unloaded prop gun | Alec Baldwin |
| 8 | Rumors of soccer player Erling Braut Haaland switching teams | Erling Braut Haaland |
| 10 | Facebook changes its name to Meta | Mark Zuckerberg |

Table A.1.: Topics and relevant entity for the summaries in Table A.2.

# A. Generated summaries

This appendix presents some of the generated summaries to (1) display how the output looks for the different systems and (2) present data that are used for discussion (e.g., time consumption). Table A.1 presents the topics of the article collections, while Table A.2 presents the summaries. Note that in both Table A.1 and Table A.2, "ID" refers to the ID of a collection, which is taken from the working dataset (Section 4.1). Thus, ID can be used to cross-reference summaries and topics. Note that the summaries were generated on a Macbook Pro 2019.

| ID | System | Generated summary | Time |
|---|---|---|---|
| 0 | Extractive | FAIRFAX, Va. (AP) — A trial over libel allegations by Johnny Depp against his ex-wife, Amber Heard, will likely turn into a six-week, mudslinging soap opera, Heard's lawyers warned a Virginia jury Tuesday. Mr. Depp's lawsuit, filed in 2019, revolves around the 2018 op-ed written by Ms. Heard titled, "I spoke up against sexual violence — and faced our culture's wrath. The op-ed does not mention Mr. Depp by name, but in it, Ms. Heard wrote that two years before the article's publication, she became a "public figure representing domestic abuse.". | 21s |
| 0 | Abstractive CNN/Daily Mail | Jury selection has begun in a libel lawsuit filed by Johnny Depp against his ex-wife, actress Amber Heard. Depp, 58, has sued Heard for $50 million, saying she defamed him when she penned a 2018 opinion piece in the Washington Post about being a survivor of domestic abuse. | 237s |
| 1 | Extractive | Musk, who calls himself a free-speech absolutist and has been critical of Twitter, disclosed a 9.1% stake on April 4 and said he plans to bring about significant improvements at the social media platform. His appointment to the board was to become effective on Saturday and would have prevented him from being a beneficial owner of more than 14.9% of common stock. Meanwhile, Twitter's board is debating using a method called a "poison pill," to prevent Musk from pushing the company into selling, two people with knowledge of the situation said, The New York Times reported. | 19s |
| 1 | Abstractive CNN/Daily Mail | Twitter's board is debating using a method called a "poison pill" to prevent Musk from pushing the company into selling. The "poison pill" will be triggered if Musk, or any other investor, expands their stake in Twitter to 15% without the board's approval. | 233s |

| | | | |
|---|---|---|---|
| 5 | Extractive | “Jokes at my expense are a part of the job, but a joke about Jada’s medical condition was too much for me to bear and I reacted emotionally. "Mr. Smith was asked to leave the ceremony and refused, we also recognize we could have handled the situation differently,” the Academy said in a statement. Rock was presenting the award for best documentary at the Oscars when Smith went on stage and slapped Rock after the comedian made a joke about Jada Pinkett Smith’s hair. | 10s |
| 5 | Abstractive CNN/Daily Mail | On Sunday evening, Rock was struck by Smith after he made a joke about the King Richard star’s wife Jada Pinkett Smith, referencing her shaved head. While Smith, 53, has since apologized for his actions, the Academy condemned his behavior and has launched a formal investigation. The Academy said disciplinary action for Smith could include suspension, expulsion or other sanctions. | 143s |
| 7 | Extractive | The bureau also documented gun safety complaints from crew members that went unheeded and said weapons specialists were not allowed to make decisions about additional safety training. Alec Baldwin in a statement through his attorney said a report released by New Mexico’s Occupational Health and Safety Bureau on Wednesday "exonerates" the actor over last October’s fatal on-set shooting. The new occupational safety report confirms that a large-caliber revolver was handed to Baldwin by an assistant director, David Halls, without consulting with on-set weapons specialists during or after the gun was loaded. | 8s |
| 7 | Abstractive CNN/Daily Mail | Baldwin was pointing a gun at cinematographer Halyna Hutchins inside a small church during setup for the filming of a scene when it went off. Hutchins, 42, died and director Joel Souza was wounded when the bullet passed through her into his shoulder. | 151s |

| | | | |
|---|---|---|---|
| 8 | Extractive | Fabrizio Romano goes on to reveal that once the Erling Haaland saga is wrapped up, one way or another, the Etihad club will then focus their attention on bringing in a new midfielder. Details of Man City's offer to Borussia Dortmund and Erling Haaland for the player's signature have been revealed. The latest transfer rumours including Erling Haaland, Kylian Mbappe, Youri Tielemans, Romelu Lukaku, James Maddison and more. | 8s |
| 8 | Abstractive CNN/Daily Mail | Borussia Dortmund talisman Erling Haaland is set to select his new employers in the upcoming two weeks. Details of Man City's offer to Borussia Dortmund and Erling Haaland for the player's signature have been revealed. Borussia Dortmund forward Erling Haaland is on his way to Manchester City. The 21-year-old Norwegian striker has clocked that United are well behind Manchester City and Real Madrid on the pitch. Manchester United reportedly have no chance of signing Erling Haaland. | 57s |
| 10 | Extractive | cambio definitivo de la compañía hacia el metaverso Ahora, Facebook se llamará Meta . "I believe the metaverse is the next chapter of the internet and it's the next chapter of our company too," Zuckerberg said Thursday. Google changed its corporate name to Alphabet to cover their other products and services, but the public will continue to associate Meta and all its services with Facebook and its founder Zuckerberg, Selepak said. | 12s |
| 10 | Abstractive CNN/Daily Mail | The name change follows Meta's disclosure on Monday that it will start breaking out financial results for the division known as Reality Labs. Mark Zuckerberg said in a presentation at Facebook's Connect conference, held virtually on Thursday. The company seems increasingly aware that it doesn't own the foundations of the digital real estate most users occupy. | 192s |

Table A.2.: Summaries generated from the working dataset.

# B. Simplified Example of the Data Preparation Flow

This appendix presents a simplified example of how the data preparation phase (Section 5.2) processes input data in Figure B.1. First, the input document(s) is divided into sentences and marked with entities mentioned in that sentence. Note that only the Wikidata ID of entities is stored along with sentences, as different mentions can refer to the same entity. Entities in the sentences are only colored to visualize the entities that have been identified by the Named Entity Recognition (NER) system. Secondly, every sentence that do not contain the input entity, which in Figure B.1 is "Doctor Strange", are removed. Thus, every remaining sentence contains a mention of the input entity.

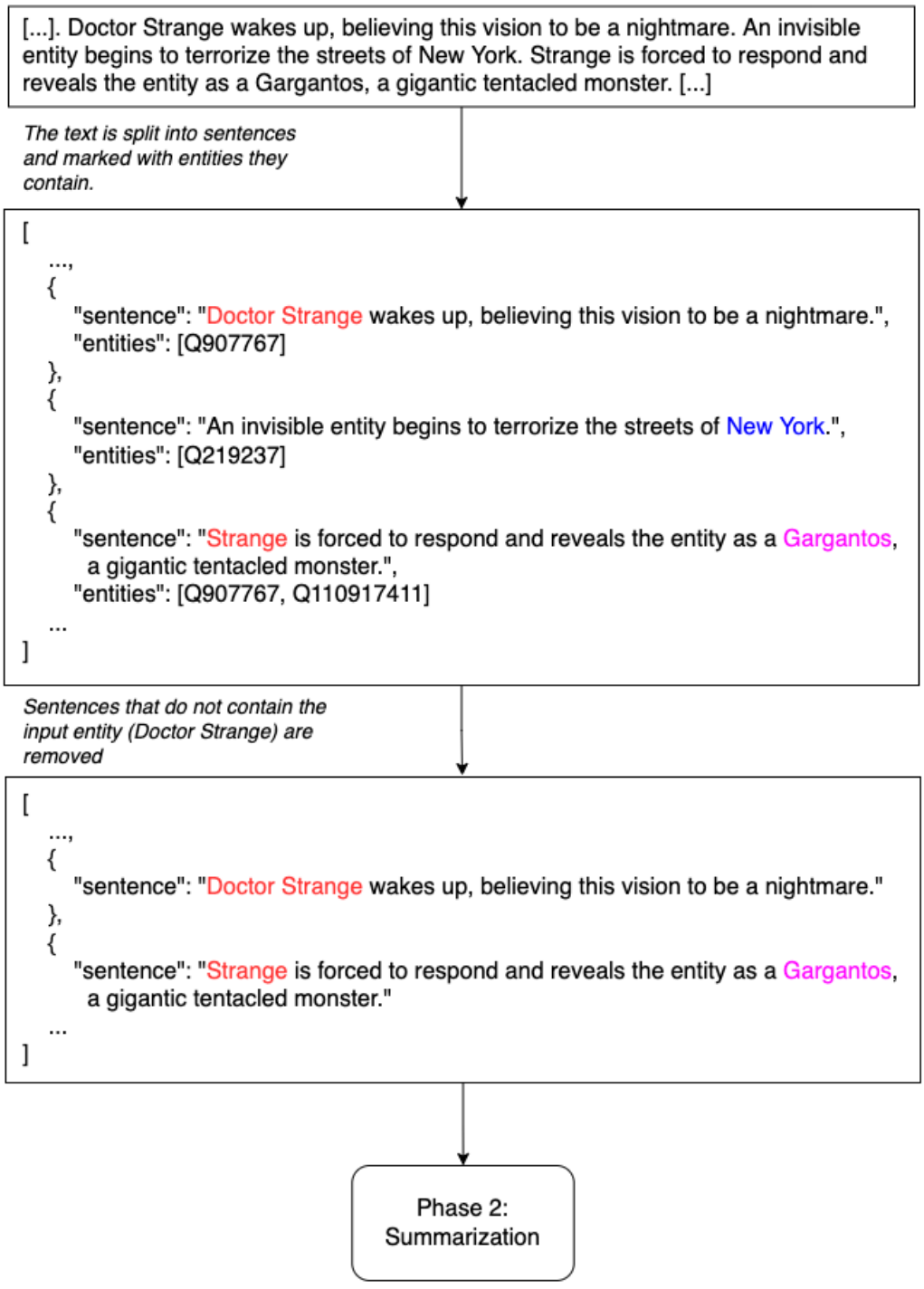



Figure B.1.: A simplified example of how input text is processed during the data preparation phase.

# C. Survey

This appendix includes the entire survey as a pdf file. The original structure of the survey, with one topic per page, was lost while converting the Google Form survey to pdf. However, all the summaries and questions are still present. The survey was conducted in Norwegian, but questions are translated to English in Section 6.3.

Note that no information about whether the summary was extractive or abstractive was given in the survey. Thus, to identify which are which, the following bullet points are intended as a guide:

- The part "1 av 4 - Johnny Depp": Summary 1 is extractive. Summary 2 is abstractive.
- The part "2 av 4 - Alec Baldwin": Summary 1 is abstractive. Summary 2 is extractive.
- The part "3 av 4 - Will Smith": Summary 1 is abstractive. Summary 2 is extractive.
- The part "4 av 4 - Erling Braut Haaland": Summary 1 is extractive. Summary 2 is abstractive.

Alternatively, the question numbering in the survey can be used. The extractive summaries are number 2, 7, 11, and 14. The abstractive summaries are number 3, 6, 10, and 15.

# Masterundersøkelse

Tusen takk for at du vil hjelpe meg med min masteroppgave 😄

Kort info om undersøkelsen:
I min masteroppgave har jeg laget to systemer som kan oppsummere en samling av nyhetsartikler. Systemene kan også knytte oppsummeringene til en person i artiklene. For eksempel kan man få en oppsummering knyttet til Trump fra artikler om stormingen av kongressen.

På hver spørsmålside vil det først bli presentert:
- Temaet i artiklene
- Hvilken person oppsummeringene knyttes til
- To oppsummeringer

Du vil så bli bedt om å svare på hvor enig du er i noen påstander knyttet til oppsummeringene.

NB:
Data fra undersøkelsen vil aldri knyttes opp mot deg personlig. Alt vil være anonymisert, samt at all data som benyttes i masteren vil være aggregert og ikke på individ-nivå. Dataen slettes umiddelbart etter jeg har tolket resultater fra de.

Lykke til 😄👍
Og igjen: tusen takk!

*Må fylles ut

1 av 4 - Johnny Depp

## Tema: Rettsak mellom Johnny Depp og Amber Heard

## Oppsummeringene er knyttet til: Johnny Depp

1. Er du kjent med temaet? *

*Markér bare én oval.*

- ◯ Ja
- ◯ Nei
- ◯ Litt

## Oppsummering 1:

2. FAIRFAX, Va. (AP) — A trial over libel allegations by Johnny Depp against his ex-wife, Amber Heard, will likely turn into a six-week, mudslinging soap opera, Heard's lawyers warned a Virginia jury Tuesday. Mr. Depp's lawsuit, filed in 2019, revolves around the 2018 op-ed written by Ms. Heard titled, “I spoke up against sexual violence — and faced our culture's wrath. The op-ed does not mention Mr. Depp by name, but in it, Ms. Heard wrote that two years before the article's publication, she became a “public figure representing domestic abuse.” *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen har variert innhold (lite repetisjon) | ◯ | ◯ | ◯ | ◯ |
| Temaet kommer tydelig fram i oppsummeringen | ◯ | ◯ | ◯ | ◯ |
| Det er tydlig at oppsummeringen handler om Johnny Depp | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen kunne vært skrevet av et menneske | ◯ | ◯ | ◯ | ◯ |

## Oppsummering 2:

3. Jury selection has begun in a libel lawsuit filed by Johnny Depp against his ex-wife, actress Amber Heard. Depp, 58, has sued Heard for $50 million, saying she defamed him when she penned a 2018 opinion piece in the Washington Post about being a survivor of domestic abuse. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen har variert innhold (lite repetisjon) | ◯ | ◯ | ◯ | ◯ |
| Temaet kommer tydelig fram i oppsummeringen | ◯ | ◯ | ◯ | ◯ |
| Det er tydlig at oppsummeringen handler om Johnny Depp | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen kunne vært skrevet av et menneske | ◯ | ◯ | ◯ | ◯ |

4. Foretrekker du oppsummering 1 eller 2? *

*Markér bare én oval.*

- ◯ Oppsummering 1
- ◯ Oppsummering 2
- ◯ Vet ikke

2 av 4 - Alec Baldwin

**Tema: Fatal ulykke på filmsett etter Alec Baldwin skøyt med rekvisittpistol**

**Oppsummeringene er knyttet til: Alec Baldwin**

5. Er du kjent med temaet? *

*Markér bare én oval.*

- Ja
- Nei
- Litt

## Oppsummering 1:

6. Baldwin was pointing a gun at cinematographer Halyna Hutchins inside a small church during setup for the filming of a scene when it went off. Hutchins, 42, died and director Joel Souza was wounded when the bullet passed through her into his shoulder. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | | | | |
| Oppsummeringen har variert innhold (lite repetisjon) | | | | |
| Temaet kommer tydelig fram i oppsummeringen | | | | |
| Det er tydlig at oppsummeringen handler om Alec Baldwin | | | | |
| Oppsummeringen kunne vært skrevet av et menneske | | | | |

## Oppsummering 2:

7. The bureau also documented gun safety complaints from crew members that went unheeded and said weapons specialists were not allowed to make decisions about additional safety training. Alec Baldwin in a statement through his attorney said a report released by New Mexico's Occupational Health and Safety Bureau on Wednesday "exonerates" the actor over last October's fatal on-set shooting. The new occupational safety report confirms that a large-caliber revolver was handed to Baldwin by an assistant director, David Halls, without consulting with on-set weapons specialists during or after the gun was loaded. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen har variert innhold (lite repetisjon) | ◯ | ◯ | ◯ | ◯ |
| Temaet kommer tydelig fram i oppsummeringen | ◯ | ◯ | ◯ | ◯ |
| Det er tydlig at oppsummeringen handler om Alec Baldwin | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen kunne vært skrevet av et menneske | ◯ | ◯ | ◯ | ◯ |

8. Foretrekker du oppsummering 1 eller 2? *

*Markér bare én oval.*

- ◯ Oppsummering 1
- ◯ Oppsummering 2
- ◯ Vet ikke

3 av 4 - Will Smith

**Tema: Will Smith slapper Chris Rock etter vits om kona**

## Oppsummeringene er knyttet til: Will Smith

9. Er du kjent med temaet? *

*Markér bare én oval.*

- Ja
- Nei
- Litt

## Oppsummering 1:

10. On Sunday evening, Rock was struck by Smith after he made a joke about the King Richard star's wife Jada Pinkett Smith, referencing her shaved head. While Smith, 53, has since apologized for his actions, the Academy condemned his behavior and has launched a formal investigation. The Academy said disciplinary action for Smith could include suspension, expulsion or other sanctions. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | | | | |
| Oppsummeringen har variert innhold (lite repetisjon) | | | | |
| Temaet kommer tydelig fram i oppsummeringen | | | | |
| Det er tydlig at oppsummeringen handler om Will Smith | | | | |
| Oppsummeringen kunne vært skrevet av et menneske | | | | |

## Oppsummering 2:

11. "Jokes at my expense are a part of the job, but a joke about Jada's medical condition was too much for me to bear and I reacted emotionally. "Mr. Smith was asked to leave the ceremony and refused, we also recognize we could have handled the situation differently," the Academy said in a statement. Rock was presenting the award for best documentary at the Oscars when Smith went on stage and slapped Rock after the comedian made a joke about Jada Pinkett Smith's hair. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen har variert innhold (lite repetisjon) | ◯ | ◯ | ◯ | ◯ |
| Temaet kommer tydelig fram i oppsummeringen | ◯ | ◯ | ◯ | ◯ |
| Det er tydlig at oppsummeringen handler om Will Smith | ◯ | ◯ | ◯ | ◯ |
| Oppsummeringen kunne vært skrevet av et menneske | ◯ | ◯ | ◯ | ◯ |

12. Foretrekker du oppsummering 1 eller 2? *

*Markér bare én oval.*

- ◯ Oppsummering 1
- ◯ Oppsummering 2
- ◯ Vet ikke

4 av 4 - Erling Braut Haaland

**Tema: Rykter sier at Erling Braut Haaland bytter klubb til Manchester City**

**Oppsummeringene er knyttet til: Erling Braut Haaland**

13. Er du kjent med temaet?

*Markér bare én oval.*

- Ja
- Nei
- Litt

## Oppsummering 1:

14. Fabrizio Romano goes on to reveal that once the Erling Haaland saga is wrapped up, one way or another, the Etihad club will then focus their attention on bringing in a new midfielder. Details of Man City's offer to Borussia Dortmund and Erling Haaland for the player's signature have been revealed. The latest transfer rumours including Erling Haaland, Kylian Mbappe, Youri Tielemans, Romelu Lukaku, James Maddison and more. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | | | | |
| Oppsummeringen har variert innhold (lite repetisjon) | | | | |
| Temaet kommer tydelig fram i oppsummeringen | | | | |
| Det er tydlig at oppsummeringen handler om Erling Braut Haaland | | | | |
| Oppsummeringen kunne vært skrevet av et menneske | | | | |

## Oppsummering 2:

15. Borussia Dortmund talisman Erling Haaland is set to select his new employers in the upcoming two weeks. Details of Man City's offer to Borussia Dortmund and Erling Haaland for the player's signature have been revealed. Borussia Dortmund forward Erling Haaland is on his way to Manchester City. The 21-year-old Norwegian striker has clocked that United are well behind Manchester City and Real Madrid on the pitch. Manchester United reportedly have no chance of signing Erling Haaland. *

*Markér bare én oval per rad*

| | Uenig | Litt uenig | Litt enig | Enig |
|---|---|---|---|---|
| Språket har god flyt og setningene henger sammen | | | | |
| Oppsummeringen har variert innhold (lite repetisjon) | | | | |
| Temaet kommer tydelig fram i oppsummeringen | | | | |
| Det er tydlig at oppsummeringen handler om Erling Braut Haaland | | | | |
| Oppsummeringen kunne vært skrevet av et menneske | | | | |

16. Foretrekker du oppsummering 1 eller 2? *

*Markér bare én oval.*

- Oppsummering 1
- Oppsummering 2
- Vet ikke

## Avsluttende spørsmål

17. Hva er ditt kjønn? *

*Markér bare én oval.*

- Kvinne
- Mann
- Annet

18. Hvor gammel er du? *

*Markér bare én oval.*

- Under 18 år
- 18-30 år
- Over 30 år

19. Jobber du mye med tekst/språk til vanlig? *

*Markér bare én oval.*

- Ja
- Nei

20. Hvordan vil du rangere dine engelskkunnskaper (1 er dårligst, 6 er best)? *

*Markér bare én oval.*

| 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| ◯ | ◯ | ◯ | ◯ | ◯ | ◯ |

21. Kunne du sett for deg å bruke et slikt oppsummeringssystem (til hva som helst)? *

*Markér bare én oval.*

- Ja
- Ja, hvis det fungerte bedre
- Nei
- Vet ikke

Martin Stiles

# Entity-Related Multi-Document News Summarization

NTNU
Norwegian University of
Science and Technology