Bauhaus-Universität Weimar
Faculty of Media
Degree Programme Computer Science and Media

Learning Unified Multi-Document Summarization From Collaborative Journalism

Master's Thesis

Yasar Naci Gündüz
Born Aug. 12, 1988 in Istanbul

Matriculation Number 114284

1. Referee: Prof. Dr. Benno Stein
2. Referee: Prof. Dr. Andreas Jakoby

Submission date: August 19, 2019

Declaration

Unless otherwise indicated in the text or references, this thesis is entirely the product of my own scholarly work.

Weimar, August 19, 2019

. . . .
Yasar Naci Gündüz

Abstract

Over the last decade, while the availability of journalistic content on the web has exploded, its popularity among consumers has sharply declined. Research has found evidence of shortening reading attention spans, and a diminishing grasp of current events, especially among younger generations. In this study, we propose to apply automatic multi-document summarization to generate short executive summaries from multiple news articles covering the same event. An automatic summarizer, in general, can employ two different construction strategies: extractive and abstractive. The former excels at collecting important facts, while the latter is better at producing a coherent final text. Our method unifies extractive and abstractive approaches to benefit from the advantages of both strategies in news summarization. In order to develop our unified extractive-abstractive summarization model, we first create a new training dataset which comprises clusters of related news articles, alongside their summaries. The dataset we collected from online collaborative journalism sites is also one of the first large-scale multi-document datasets in the news domain. Based on these data, we develop an extractive summarizer to gather salient sentences from input news articles, and an abstractive summarizer that can weave these sentences into a coherent and non-redundant narrative. In a comprehensive evaluation study, we combine automatic and manual evaluation techniques to investigate the quality of our unified summarization system; our results show that including the extractive stage in the summarization process improves both the content selection and readability of the final abstractive summaries. While our error analysis finds systematic errors still present in the final summaries, our results do make a contribution towards helping news consumers better navigate the modern information landscape.

Contents

1 Introduction 1

2 Background and Related Work 4
2.1 Fundamentals of Automatic Summarization 4
2.1.1 Extractive Summarization 5
2.1.2 Abstractive Summarization 6
2.1.3 Neural Abstractive Summarization 8
2.2 Existing Approaches for Multi-Document Summarization 9
2.2.1 The Evolution of Extractive Multi-Document Summarization 10
2.2.2 The Evolution of Abstractive Multi-Document Summarization 12

3 Dataset 14
3.1 Online Collaborative Journalism: Wikinews 14
3.2 Dataset Construction 16
3.2.1 Processing English Wikinews Dumps 17
3.2.2 English Wikipedia Dump Files 17
3.3 Retrieving the Sources from the Web 18
3.3.1 Python urllib and Wayback Machine 18
3.3.2 Main Content Extraction 19
3.3.3 Detecting Language 20
3.4 Structure of the Data Storage 20

4 Learning Multi-Document News Summarization 22
4.1 Extracting Salient Sentences 23
4.1.1 Sentence Scoring 23
4.1.2 Strategies for Affordable Input Length: Diversity and Completeness 25
4.1.3 Removing Redundancy 26
4.1.4 Evaluating the Sentence Extraction 34

4.2 Creating the Narrative 38
4.2.1 Neural Abstractive Summarization 38
4.3 Experiments and Evaluation 41
4.3.1 Training Models 41
4.3.2 Automatic Evaluation for Content 42
4.3.3 Manual Evaluation for Readability 43
4.3.4 Observations on the Final Summaries 48

5 Conclusion 52

A Sample XML entry for the news page from English Wikinews Dump file 2017 55

B Code snippet to generate JSON and XML output 58

C Miscellaneous Information 60
C.1 Namespaces 60
C.2 Full list of the News Websites from allsides data collect 61
C.3 Directory Structure 62

D CST Duplication Filter 63
D.1 ICSIBOOST sample data file 63
D.1.1 A snippet from milan9-gulfai-binary.data 63
D.2 ICSIBOOST names file 63
D.2.1 milan9-gulfair-binary.names 63
D.3 Trained data for ICSIBoost(shyp file) 64
D.3.1 A snippet from milan9-gulfair-binary.shyp 64
D.4 CST Bank Data Samples 65
D.4.1 CST Bank sample article in XML format (Cluster: Milan9 File: 1.docsent) 65

Bibliography 67

Acknowledgements

First and foremost, I would like to thank Michael Völske and Wei-Fan Chen. This work would not be possible without their unwavering guidance and dedicated support.

I would like to extend my sincere thanks to Prof. Dr. Benno Stein for accepting my work under his supervision.

Last but by no means least, my sincere thanks go to my friends Endre Papp, Hatice Dayıoğlu, Seymur Mammadrza, Büşra Torgay, Özgür Dönmezer, and Alp Kayalar for never hesitating to help me in time of need.

Chapter 1

Introduction

The main goal of this thesis is improving the quality of summarization for news articles by employing multi-document summarization techniques. A multi-document summarization system is a machine learning procedure aimed at creating a brief outline of the text gathered from multiple documents about the same topic. A sufficient multi-document summarization system not only shortens the source text, but also creates a summary which preserves the meaning of the articles, includes the key points, and forms a seamless narrative while doing so. To understand the idea behind this study, one should understand why summarizing the news is important, what is essential to generate a proper summary and how our proposal of a unified summarization method is relevant.

It is not very difficult to find an article about the death of print media and the transformation of journalism giants into online services, but there also exists the fact that traditional journalism and media is getting less and less popular whether it is online or not. As reported by CNBC News (Wastler [2013]), half of their readers cease to read after the first three paragraphs, and very few read until the end. The unpopularity of traditional journalism is even more common amongst the young generation, which points out that the situation might get even more severe in the future. According to Rogers [2017], a survey by Pew Research Center showed that the participants belonging to the youngest age group (18-34) are the least informed about current events. Americans aged 18-34 are less informed than Americans aged 34-49, and those in turn are less informed than Americans above age 50. According to the book *The Dumbest Generation* written by English professor at Emory University Mark Bauerlein (Bauerlein [2008]), mentioned in the same article, interests of the younger generation are different than what newspapers can offer. After years of success in knowledge sharing, people eventually lost interest in mainstream journalism. However, considering the information pollution on the internet (Pandita [2014]), newspapers are still one of the most reliable sources of information. Creating less time consuming and more engaging content might help newspapers to regain their former popularity. However, creating such content would be very costly for humans, considering it must be done every day. On the other hand, an automatic system can generate this content for far less cost once it is created. Even though a stand-alone summarization system could not solve the issue altogether, we argue that it is potentially an essential part of producing appealing content to attract people to read information from reliable sources.

The key to a good summary is a thorough analysis of the text to be summarized. The American Press Institute describes journalism as "*the activity of gathering, assessing, creating, and presenting news and information.*"[1] The final transcript must be an amalgam which contains all the prominent information from different sources to be informative, coherent and concise to hold the reader's attention until the very end. Another aspect of summarization and writing, in general, is the narration. Bill Kovach and Tom Rosenstiel's definition of journalism is "*the storytelling with a purpose*" in their book *The Elements of Journalism* (Kovach and Rosenstiel [2014]). A good story not only contains all the facts and aspects but also presents the information with an appropriate narrative in the course of engaging the audience. Hence, in this study, we investigated automatic summarization methods to develop a multi-document unified summarizer, which unifies two construction methods: *extractive and abstractive* (Othman et al. [2014]). By doing so, we aimed to handle content selection with extractive summarization and generate the summaries with an appropriate narrative by using abstractive summarization. Furthermore, several studies (Suanmali and Salim [2008], Sakhare and Rajkumar [2014]) showed that most of current multi-document summarization models rely on either an abstractive or an extractive approach. While a few previous approaches do combine both, most of those techniques still do not include neural abstractive summarization models, which exhibited significant success in creating well-narrated summaries and became very popular in recent years (Shi et al. [2018]). In our model, we adapt a neural abstractive method developed for single-document summarization to the multi-document setting. In order to train the neural abstractive summarizer with a multi-document clustered dataset, we developed an extractive summarizer. Therefore, our unified method relies on extractive summarization not only for better content selection but also converting multi-document clusters to single-document training data.

The remainder of this work is structured as follows: Chapter 2 introduces the fundamentals of automatic summarization systems, and reviews previous

[1] `https://www.americanpressinstitute.org/journalism-essentials/what-is-journalism/`

work on multi-document summarization, which addresses the problem of producing one coherent summary for a *cluster* of related input documents. Even though supervised summarization systems such as ours vastly depend on training data, previous datasets for multi-document tasks contain fewer than 100 document clusters (Cao et al. [2015]). To provide a more extensive dataset, we collected news articles and their respective summaries created collaboratively. As a result, we created one of the first large-scale multi-document summarization datasets in the news domain, which consists of 11,688 clusters and 39,121 documents. In Chapter 3, we reveal the structure, construction and retrieval strategies of the dataset. In Chapter 4, firstly, we introduce the extractive summarization method we reconstructed alongside the mechanism that we implemented to remove duplicate information. We also examined the best strategy for producing extractive summaries to train the abstractive summarizer amongst the strategies we proposed. Next, we discuss the abstractive summarization method that we adopted and the experiments we conducted to examine the quality of our approach, both in terms of content selection, and in terms of readability. While the former can be evaluated with automatic measures, the latter required an extensive study involving human annotators. In the process, we documented key observations on the limitations of non-expert annotators for summary quality. While our observations on the final summaries show that the framework we proposed still needs improvement, evaluation results revealed that combining extractive and abstractive methods showed certain improvement over a method using only an abstractive summarizer both in content selection and readability. Finally, in Chapter 5, we discussed the possibilities for future work regarding the improvements, drawbacks, and failures of our approach.

Chapter 2

Background and Related Work

A summary is a brief statement or account of the main points of a certain topic[1]. A good summary includes the most important parts of the text and excludes the redundant details. With the extraordinary increase of the data during the last decades, summarization becomes a more and more reasonable answer for the demands of the users for the text data which is less time consuming and yet still adequately informing.

2.1 Fundamentals of Automatic Summarization

To get a better understanding of automatic summarizers, first, we need to understand its fundamentals and cornerstones. Automatic summarizers can serve various purposes from text simplification (Margarido et al. [2008]) to generating Wikipedia pages (Liu et al. [2018]). During decades of research, numerous approaches have shown to be successful and established a basement for many to come. Othman et al. [2014] suggested five main categories for the taxonomy of text summarization: *number of sources, summary construction method, summary target, information content and approach.* In this and the following sections, we mention the approaches in the categories which are important for this study.

Even though automatic summarizers can be categorized with the types presented in Figure 2.1, they are generally divided into two main categories under the summary construction methods: *Extractive* and *Abstractive* (Khan and Salim [2014]). The number of sources also can be deemed as another significant distinction since the multi-document approaches have been evolved from single-document ones. Thus in this section, we introduce the fundamentals of summarization with single-document approaches for both construction

[1] https://en.oxforddictionaries.com/definition/summary

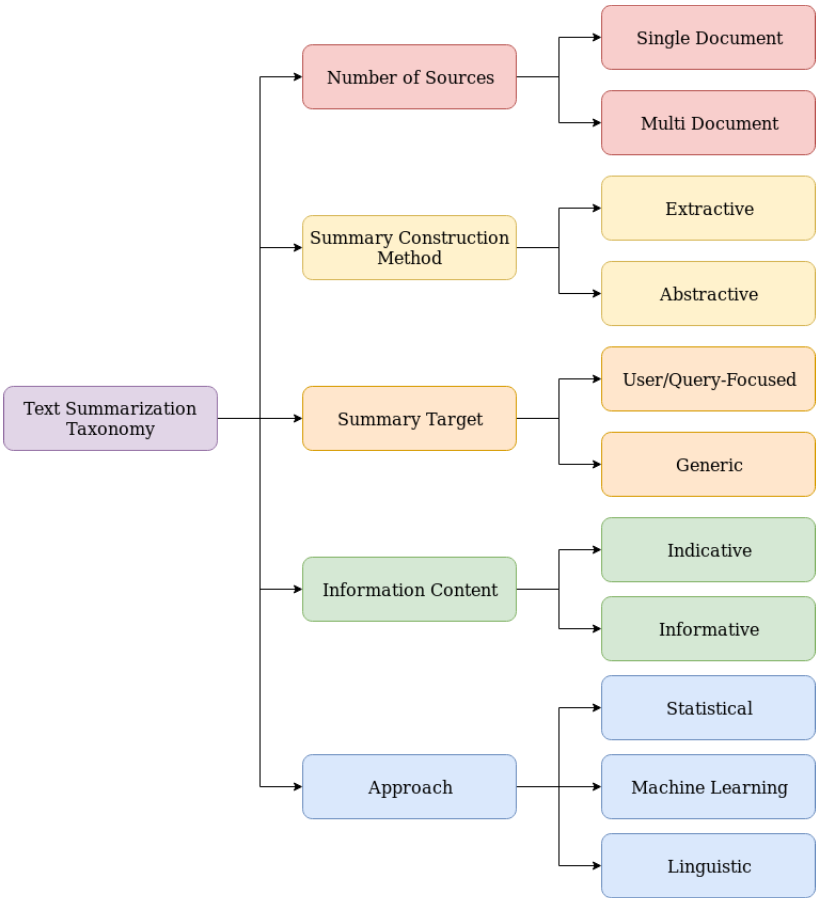

Figure 2.1: System design decisions for automatic text summarization (Othman et al. [2014])

methods, and we proceed to the multi-document approaches in the following section.

2.1.1 Extractive Summarization

Even though the algorithms used vary and differ, according to Nenkova and McKeown [2012], extractive summarization can be represented by three semi-independent tasks: *intermediate representation, sentence scoring and sentence selection.*

Intermediate Representation

Intermediate representations are the starting point of computer understanding in extractive summarization. Important sections of the content are pinpointed based on this representation. Depending on the concerns of the task, different approaches can be used.

Topic representation approaches interpret topics in the text to provide an intermediate representation. In *lexical chain approaches* a thesaurus is used to detect the topics or the concepts of the semantically related words and assign weights to the concepts; in *latent semantic analysis* word co-occurrence patterns are interpreted as topics as well as the weights for each pattern; in *Bayesian topic models* the input is treated as a mixture of topics, and for each topic there is a table of word probabilities assigned.

Indicator representation approaches, on the other hand, convert each sentence into a list of indicators. The importance of the indicators is determined by the metrics such as sentence length, sentence position, or presence of certain terms or phrases. In *graph models*, the whole document is represented as a network in which every sentence is related to the others.

Sentence Scoring

Following the development of intermediate representation, a score that indicates the importance is assigned for each sentence in the document. In topic representation approaches, the score usually corresponds to the expressiveness of the sentence for the topic and the capability of including the information about different topics. In indicator representation approaches, the weight is commonly determined by the evidence from the different indicators.

Sentence Selection

At the very last step, there are also several different approaches that developers can employ to choose which sentences make to the final summary. The *best n* approach chooses the first n results from a list of sentences ordered descendingly by their scores. *Maximal marginal relevance* approaches use an iterative greedy procedure for the selection. This procedure recalculates the scores after each selection based on its similarity with the selected sentences. The sentences with higher similarity get the lower scores. *Global selection* approaches make the selection with constraints which maximizing overall importance and coherence as well as minimizing the redundancy.

2.1.2 Abstractive Summarization

Early approaches to abstractive summarization techniques can be generally classified under two categories: *structure-based approaches* and *semantic-based approaches* (Kasture et al. [2014]). The methods falling under the *structure-based approaches* are *tree based*, *template based*, *ontology-based*, *lead and body* *phrase* and *rule-based* methods. The methods falling under *semantic-based approaches* are *multimodal semantic model*, *information item based model* and *semantic graph-based model* (Khan and Salim [2014]).

Structure Based Methods

Structure based methods work through cognitive schemas such as templates or extraction rules to encode the most important information in the given text (Khan and Salim [2014]).

In *tree based methods*, the content is represented in the form of a dependency tree. A shallow parser does the pre-processing, and then the sentences are mapped to the predicate-argument structure. Different algorithms can be used for sentence selection. In the end, a language generator is applied to generate the final summary.

In *template based methods*, a template represents the document. The technique uses linguistic patterns or extraction rules to map the fragments of the text to the template slots.

In *ontology based methods*, the experts defines the domains of ontology. Following the initial domain production, the method identifies and classifies the meaningful terms and generates a membership degree for each term by fuzzy interference.

In *lead and body phrase methods*, the phrases of the same syntactic head chunk from the lead and body sentences are included or replaced to rewrite the lead sentences.

In *rule based methods*, documents are represented in terms of categories. The content selection module forms the questions based on categories, generates extraction rules depending on the questions and selects the best candidates by using the rules. Finally, the summary is generated based on the generation patterns.

Semantic Based Methods

Semantic based methods exploits the semantic structure of the document to feed the *natural language generation* (NLG) systems. Methods process the linguistic data to find noun and verb phrases (Khan and Salim [2014]).

Multimodal semantic model captures the concepts and relations among them, rates the concepts based on their information density and uses the rated concepts to generate summary sentences (Kasture et al. [2014]).

Information item based model follows a different path to produce a summary. Instead of producing an abstract from the input, it generates the summary from an abstract representation of the document called *information item*, which is the smallest element of information in the text (Kasture et al. [2014]).

Semantic graph based model produces a semantic graph called *rich semantic graph* (RSG), reduces the RSG and generates the abstractive summary from the reduced graph (Kasture et al. [2014]).

2.1.3 Neural Abstractive Summarization

In recent years, researchers started to incline toward neural abstractive summarization. The success of the neural attention model for abstractive sentence summarization (Rush et al. [2015]) encouraged the researchers to adopt sequence to sequence model alongside other approaches using deep learning (Shi et al. [2018]). For the very same reason, we also decided to use a novel method based on the neural abstractive summarization: *Pointer-Generator Network* (See et al. [2017]).

Before introducing some of the important earlier work that provided a basis for Pointer-Generator Network (Shi et al. [2018]), we provide some background information about Recurrent Neural Networks (RNN) since it is one of the crucial components of neural abstractive approaches.

Recurrent Neural Networks

Understanding and producing a text require to consider the input as a whole. In the human understanding of language, the semantic meaning of the words in a sentence is related to the words come before (Young et al. [2018]). Recurrent Neural Networks can manage to understand such formation due to their ability to process sequential data. This ability comes from its structure, which allows passing the information to the unit in the next time step. By doing this, RNNs can process a sequence of dependent information (i.e. words of a sentence) across time while it preserves the relationship among them.

To get a closer look and better understanding, let us unfold an RNN. In Figure 2.2 one can see the unfolding of an RNN with recurrent connections between hidden units. In this architecture, each member of the hidden state is producing output using the same function with previous ones, and each member of the hidden state is a function of the previous members. In other words, the current state of the network is dependent on the previous ones (Goodfellow et al. [2016]).

Early Work on Neural Abstractive Summarization

Rush et al. [2015] developed the first abstractive summarization approach implementing the neural networks. They proposed a fully data-driven approach and introduced a sequence-to-sequence model based on an encoder-decoder architecture for the abstractive sentence summarization. An attention-based

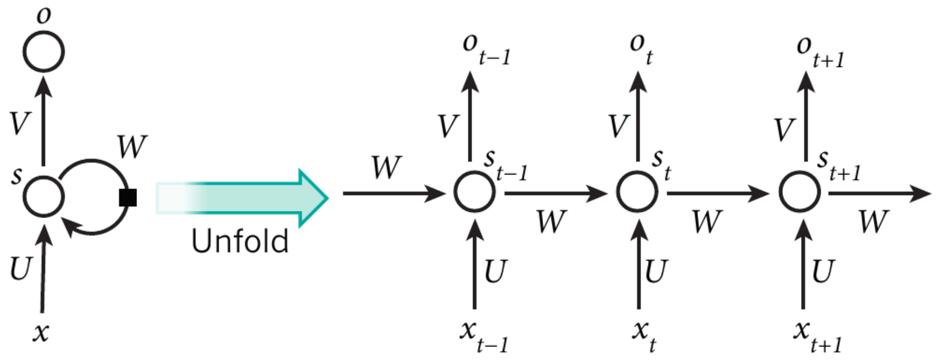

Figure 2.2: A recurrent neural network and the unfolding in time of the computation. (LeCun et al. [2015])

encoder and a decoder using the neural network language model (NNLM) that they used outperformed the traditional methods. In further work Chopra et al. [2016] replaced the feed-forward NNLM with RNN and introduced a conditional recurrent neural network. Their model showed significant improvement on the Gigaword corpus and performed competitively on DUC-2004. Nallapati et al. [2016] proposed the following novel models, each addressing different flaws of the basic encoder-decoder RNN:

1. Feature-rich encoder to capturing keywords
2. Switching generator-pointer for modelling out-of-vocabulary words
3. The hierarchical attention to capturing hierarchical document structures

Their work also provided a new dataset for abstractive summarization into multiple sentences.

2.2 Existing Approaches for Multi-Document Summarization

Through the years, researchers changed the course of text summarization from single document to multi-document summarization. Even though both types have certain differences, they also show certain similarities in terms of understanding the data and writing the summary. Just like single-document techniques, multi-document summarization also includes techniques for extractive and abstractive approaches. In Subsection 2.2.1 and Subsection 2.2.2, we introduce some important methods presented in Suanmali and Salim [2008] and Sakhare and Rajkumar [2014] for both approaches to help the readers to understand why we wanted to research a unified method.

2.2.1 The Evolution of Extractive Multi-Document Summarization

McKeown and Radev [1995] created SUMMONS which generates summaries from several documents of same or relevant topics. They improved their work in 1998 (Radev and McKeown [1998]) in a way that summarizer selects the information based on its frequency of appearance in different documents.

McKeown et al. [2001] proposed a multi-document summarization system which changes the summarization strategy dependent on the document type.

In the system proposed by Fukumoto [2004], their framework classifies the document sets into three types: a set of documents of the same topic, a set of documents of the same event type and a set of documents of the related events. The system determines the type by using information of high frequency nouns and named entities, and assigns an appropriate summarization technique based on the type of document set. After every document in the set is summarized by the assigned technique, the final summary is completed by removing the unnecessary parts.

Zhang et al. [2005] proposed a new approach for multi-document extractive summarization under the hub-authority framework where cue phrases, sentence lengths and the first sentence of the content are combined with the text content. In this approach, the sub-topics are explored by using the sub-topic features in a graph-based sentence ranking algorithm and ordered by the Markov Model. The summarizer generates the summary from the ranked sentences of certain sub-topic required by the user.

The algorithm developed by Chen et al. [2005] pre-processes the content in a way to remove redundancies and preserve the differences, then constructs a lexical chain to identify the strong chains. Based on lexical chains, it extracts the important sentences from each document and generates the summary in chronological order.

Liu et al. [2006] proposed a cluster-based method for Chinese multi-document summarization consists of sentence clustering and sentence selection.

Schlesinger et al. [2008] proposed a multilingual summarization system using CLASSY (Clustering, Linguistics, And Statistics for Summarization Yield) system architecture which uses linguistic trimming and statistical methods to produce generic or query/topic driven summaries. The architecture consists of five steps: document preparation, sentence trimming, sentence scoring, redundancy reduction, and sentence ordering.

Table 2.1 exposes the strengths and weaknesses of every technique alongside respective researchers, year of publishing, languages, and techniques used.

Table 2.1: List of extractive multi-document summarization techniques presented in Suanmali and Salim [2008] with respective researchers, year of publishing, languages, techniques used, strengths and weaknesses.

Researcher(s), Year, Reference	**Language(s)**	**Technique(s)**	**Strength(s) and Weakness(es)**
McKeown, Barzilay, Evans, Hatzivassiloglou, Teufel,Yen Kan, and Schiffman 2001	Mono-Lingual, English	MultiGen, DEMS	-Columbia system did well on grammatically but did not fare as well on cohesion and organization
Jun ichi Fukumoto 2004	Mono-Lingual, Japanese	Statistics (frequency noun, name entities)	-Mechanism of document set classification does not work well.
Junlin Zhang., Le Sun. and Quan Zhou 2005	Mono-Lingual, English	Statistics, Graph base sentence ranking	
Yan-Min Chen, Xiao-Long Wang, and Bing-Quan Liu. 2005	Mono-Lingual, Chinese	Lexical chain	-Lexical chains are effective for Chinese texts summarization
De-Xi Liu, Yan-Xiang Hi.,Dong-Hong Ji. and Hua Yang. 2006	Mono-Lingual, Chinese	Cluster-based method	-Improve the performance of Chinese multi-document summarization
Judith D. Schlesinger, Dianne P. O'Leary, and John M. Conroy. 2008	Multi-Lingual, Arabic, Englishl	Clustering, Linguistics, Statistics	-Very good summary for English document -Quality of summary depend on quality of machine translation

2.2.2 The Evolution of Abstractive Multi-Document Summarization

The work of Lloret and Sanz [2011] was centring around determining most suitable sentences generated by a word graph-based method for producing abstracts. Furthermore, they developed an extractive summarization technique to decide which abstractive sentences are more relevant and therefore, included them in the summary. Their preliminary experiments showed that a combination of either approach is an effective strategy.

Cheung and Penn [2013] proposed that better usage of the domain of the source text might enhance the quality of the abstractive summarization. They experimented with human-written ground-truth summaries, which showed (1) ground-truth summaries were more abstractive, and sentence aggregation is used more, (2) ground-truth summaries include fewer topical case frames, (3) ground-truth summaries cannot be reconstructed merely by the source text, but the usage of the documents of the same domain might make it possible. Results showed the convenience and usefulness of domain knowledge in abstractive summarization.

Genest and Lapalme [2012] developed a framework that starts with information extraction, then performs sentence selection based on a statistical approach and finally generates the summary with *natural language generation*. Using the extractive approach for concatenating and abstractive approach for paraphrasing exhibited in the test results that a combination of the methods is an effective way to produce multi-document abstractive summaries.

Liu and Liu [2009] tried to apply several sentence compression techniques to extractive summaries to produce an abstractive summary. Evaluation results showed that compression techniques applied on extractive summaries outperformed the techniques only using the abstractive summarization; however, the improvement was so small that it suggested that language generation is needed for abstractive summarization.

Honarpisheh et al. [2008] have proposed to use singular value decomposition and hierarchical clustering for developing a multi-document and multi-language summarizer. Their work relies on two resources regardless of the language: a word segmentation system and a dictionary of words along with its document frequency. They achieved to test their system successfully on Persian documents.

Goldstein et al. [2000] have discussed a multi-document extractive summarization approach that builds on single document methods exploiting available document-set information and relations between documents.

Ji et al. [2013] have examined the effectiveness of cross-document information extraction (IE) techniques on multi-document summarization. They analyzed the advantages and disadvantages of several IE-based summarization approaches, one of which showed improvement in content quality and readability.

Chapter 3

Dataset

One of the most serious shortages of supervised summarization systems is the human-generated summaries for training. In case of multi-document summarization task, the most widely-used datasets produced by Document Understanding Conference (DUC)[1] does not contain more than 100 clusters and 600 documents each (Cao et al. [2015]). Therefore, as the first contribution of this study, we decided to create a new dataset for developing our summarizer. The *Webis-wikinews-corpus* that we created for this purpose is one of the first large-scale multi-document summarization datasets in the news domain. We acquired a dataset consisting of 11,688 documents from the Wikinews/Wikipedia pages and 39,121 source articles associated with the respective main document. In Table 3.1 we provide the number of clusters and documents that our dataset contains and the datasets published by DUC in 2001, 2002 and 2004. Moreover, the following sections describe how the corpus is retrieved and constructed to create a large set of news articles -summarized or otherwise- which provides training and evaluation data to a multi-document summarization system.

3.1 Online Collaborative Journalism: Wikinews

Wikinews has been founded by the Wikimedia Foundation to create a collection of free, transparent and open source news (Weiss [2005]). Wikinews states its mission as "*To present up-to-date, relevant, newsworthy and entertaining content without bias*" (Wikimedia [2015]). The Wikinews community collaboratively produces the content to meet the requirements of this mission. According to the founder of Wikimedia, Jim Wales, Wikinews content must be written in the format of a news story. Moreover, Wikinews has content creation rules very similar to those of its sister project Wikipedia. Each story can be

[1] `http://duc.nist.gov/`

Table 3.1: The number clusters and documents of Webis-wikinews-corpus and corpora published by DUC (Cao et al. [2016])

Data Source	Cluster	Documents
DUC 2001	30	309
DUC 2002	59	567
DUC 2004	50	500
DUC Total	139	1,376
Wikinews	9,514	21,314
Wikipedia	2,174	17,807
Webis-wikinews-corpus total	11,688	39,121

edited, corrected and extended by any reporter, changes should be recorded, and sources must be present (Stuff [2004]). Wikinews reached a thousand articles almost half a year after it had been founded, and had collected more than ten thousand news articles when it was celebrating its third year online (Wikipedia contributors [2005]). It showed consistent growth over time (Figure 3.1), and by doing so, it showed an online collaborative journalism project can be an alternative to traditional journalism.

There are, of course, many people who think that reliability in journalism is difficult to handle with online and collaborative methods. The credibility of the project became a target of criticism and scepticism due to its potential incapability of having a neutral point of view and including reliable information from verified sources (Weiss [2005]).

Regardless of whether or not Wikinews is a reliable news outlet, its data is of interest for training machine learning systems. In the case of our study, Wikinews pages provide the ground-truth summaries, titles and source articles, alongside auxiliary information such as categories, reporters, or publishing dates. To construct the Webis-wikinews-corpus out of the English files provided by Wikimedia, firstly, we processed the files in a way to keep only the information that we considered to be useful. As a result of this first step, we created a JSON and an XML file containing the Wikinews/Wikipedia articles that we planned to use as a ground-truth summary, auxiliary information, and the links for the news articles cited in Wikinews/Wikipedia page. Later we iterated through the JSON file, and we acquired the main content of the source articles for every ground-truth summary. At every iteration, we saved *txt* files for ground-truth summaries, source articles and their respective auxiliary information under a specific folder structure for clustering the relevant information.

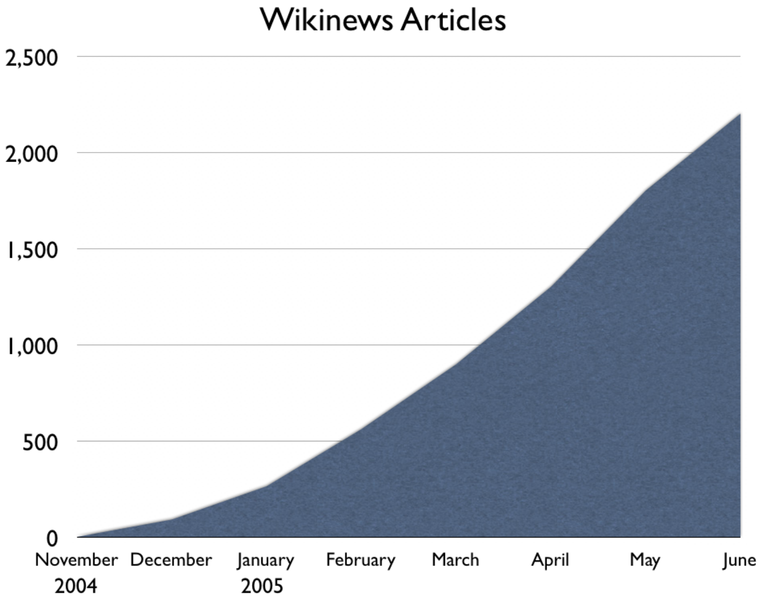

Figure 3.1: Amount of Wikinews Articles by Months (Wikipedia contributors [2005])

3.2 Dataset Construction

In keeping with the aim of creating content freely available for everyone, Wikimedia is sharing its content in XML format. All content which has been aggregated and generated by Wikimedia projects can be found in Wikidump files under the GNU Free Documentation License (GFDL) and the Creative Commons Attribution-Share-Alike 3.0 License unless it is stated otherwise[2]. It is possible to download complete copies of all Wikimedia wikis, in the form of wikitext source and metadata embedded in XML[3]. We use two of these wikis in this study: Wikinews, and the pages from Wikipedia, which contains sources to the news articles. Since the dump files contain lots of information that is beyond the scope of training a summarization system such as disclaimers or copyright notices, we used *Wikiextractor*[4] to extract the useful information out of the dump files.

[2] https://dumps.wikimedia.org/legal.html

[3] https://dumps.wikimedia.org/backup-index.html

[4] https://github.com/attardi/wikiextractor

3.2.1 Processing English Wikinews Dumps

English Wikinews dump files contain all the pages in the Wikinews with an associated id, namespace, title and miscellaneous information such as timestamp, parent id and contributor. The dump files shared on 20.10.2017 that we used in this study include 24 different namespaces (See appendix C.1). For this study, we only needed the pages which are indexed with namespace Id 0 since they are the only pages that contain news stories. The main content of the page can be found in between the XML tags with parameters: <text xml:space="preserve">, yet the content is not plain text but a text containing further information which is encoded in Wikimedia's markup style. In this encoded info, one can find useful information like linked pages, categories, sources and external links (See appendix A).

Since the data in the Wikinews dump file is far broader than we needed and encoded in a special format, a simple XML reader was not sufficient and a more specific framework was necessary. Therefore, we decided to use Wikiextractor, which we extended such that it can convert dump files into a new JSON based format that is more suitable for our project. Since the Wikiextractor was sufficient for extraction and conversion, we kept the modification limited with the parameters of the Input and Output. On the input's side, we limited the pages to be processed with the pages with namespace id 0 and on the output's side, we set the output in a way that it contains only id, URL, title, text, sources, externals, categories and reporters. For possible further usage, we also produced the same output in XML format. In section 3.3, we describe how we exploited the JSON file to construct the corpus. The modified code to create JSON and XML output can be found in Appendix B.

3.2.2 English Wikipedia Dump Files

The content of the Wikipedia pages is not a perfect choice for a ground-truth summary in our study. It is too long and it includes so many non-news sources. Hence, we decided to include only the pages which are linked to the news sources to form a dataset which is content-wise consistent and as extensive as possible at the same time.

However, the sentences linked to the news articles were scattered throughout the text. Stitching them together could form a ground-truth summary consists of irrelevant sentences, and using them separately would create a dataset for single-document summarization. Therefore, we decided to exclude the Wikipedia documents from the dataset we used for multi-document unified summarization.

Even though we could not use the documents and source articles from

Wikipedia, we kept them in the Webis-wikinews-corpus since the data can be useful in further research.

3.3 Retrieving the Sources from the Web

The data we acquired up to this point only consist of the Wikinews/Wikipedia articles, initiated metadata, and the links to the sources. However, since the system still needs the source articles to be summarized for training and testing, we developed a web scraper which is tailored for the requirements of this study. The web scraper gets the source links associated with the ground-truth summary in the JSON file, and for each URL, it performs the following tasks in the given order:

1. Download the HTML content of the page
2. If the page is unavailable, retrieve the most up-to-date copy from the *Internet Archive*
3. Extract the main content
4. Detect the language, exclude the sources which are not written in English

3.3.1 Python urllib and Wayback Machine

We used Python's urllib library, which allows executing HTTP requests by using HTTP1/1 and returns an HTTP response containing the main content of the web page[5]. Since the Wikimedia is sharing its complete content, it includes numerous source-links which belong to more than a decade ago. Many of which are broken, directed or unavailable. Thus, we decided to use the *Wayback Machine.* The Wayback Machine[6] is a digital archive of the web sites which has been launched by The Internet Archive in 2001. Ever since the project allows the users to save the website into the archive and recall all the versions saved in the system. Furthermore, The Internet Archive provides an API for those who want to use the archive. The API works with a request in the following format:

```
http://archive.org/wayback/available?url=example.com
```

,and returns the the following response containing a link to the version/versions of the page in the archive if the page has been archived[7]:

[5] https://docs.python.org/3/library/urllib.html

[6] https://en.wikipedia.org/wiki/Wayback_Machine

[7] https://archive.org/help/wayback_api.php

```
{
    "archived_snapshots": {
        "closest": {
            "available": true,
      "url": "http://web.archive.org/web/20130919044612/http://example.com/",
            "timestamp": "20130919044612",
            "status": "200"
        }
    }
}
```

Even though the Wayback Machine did not archive all the unavailable pages that we needed, it helped us to collect the most up-to-date available version of a considerable amount of pages.

3.3.2 Main Content Extraction

Text in an HTML page is consist of many parts from hyperlinks to user comments, all of which are divided and placed on the page by HTML tags. We exploited the HTML and DOM structure of the page to extract the main content. For many news website, HTML elements containing the main content has the same class name.[8]

```
<div class="zn-body__paragraph speakable">"(Kim) did not have a backup plan," a source familiar with the denuclearization talks tells CNN. "He came to Hanoi very confident and fully expecting a declaration to be signed."</div>
<div class="zn-body__paragraph speakable">But as he departed Saturday, waving to crowds before boarding that train empty handed, the cracks in that confidence had been exposed.</div>
```

Beautifulsoup library provides pythonic idioms for iterating, searching and modifying the tree that is parsed from HTML which are built upon an HTML parser[9]. Thus, we used the library to conduct an iterative search on the page and find the HTML elements containing the paragraphs belong to the article. As we mentioned before, websites of certain news agencies have a specific class name for the chunks containing the main content. A python project named *allsides-data-collect*[10] has an implemented extractor for the main content of the pages from plenty of major news agencies (See appendix C.2). Even though we covered a good portion of the news sources from Wikinews dumps by using

[8] HTML snippet from CNN news website: https://edition.cnn.com/2019/03/02/politics/trump-kim-summit-dream-ripley-intl/index.html

[9] https://pypi.org/project/beautifulsoup4/

[10] https://git.webis.de/webisstud/allsides-data-collect-patrick

the codes from allsides-data-collect, rest of the pages still needed to be scraped to produce a training data as extensive as possible. Thus, we implemented an algorithm which is able to extract the main content of any page. To do so, we exploited the DOM structure. In an HTML page usually, elements are nested into each other, creating a tree structure; therefore, the different type of elements on the page has different depths on the DOM tree. We began with clustering the elements of the page by their depth. Typically the elements containing the paragraphs of the news article has the longest text content. Hence, we extracted the combined content of the elements of the same depth, which contains the longest text.

3.3.3 Detecting Language

English Wikinews dumps are consist of the articles written in English however since the Wikinews is welcoming the contribution of any writer from any country, it was not guaranteed that all of the sources are from the newspapers which publish only in English.This study, on the other hand, merely focuses on summarizing the articles in English; therefore, any resource which contains an article written in any other language had to be filtered out. For this purpose we decided to use Google's language-detection library [11]. After the extraction of the main content for every each URL, the system determines the content's language and proceeds to the next URL without saving if the content is not in English.

3.4 Structure of the Data Storage

We argue that a good data storage not only contains the necessary information but also provides easy access to the data. The corpus we created in this study is saved in a specific structure which allows the programmers to access a certain type of documents such as ground-truth summaries, source articles, metadata by fetching certain prefixes or postfixes through the iterations. Every single Wikinews or Wikipedia page retrieved from the dumps are saved in separated folders, names of which start with a prefix stating the type of the wiki and continues with the id of the page in the dump files joined with "_". If the article is from Wikinews, folder names start with "wn" and if it is from a Wikipedia page "wd". Contents of the pages are placed in the folders in txt format with the same name alongside the folders containing the contents of the sources. Source folders are named by the similar convention which has a "_res_" tag in between wiki-type and id, and ends with an "_" followed by

[11] https://code.google.com/archive/p/language-detection/

the ordinals. Content of the sources can be found in the source folders in a txt file of the same name. As mentioned before, the Webis-wiki-corpus also provides users with auxiliary information. For each main folder, there exists a txt file for categories which starts with the main folder name and ends with "_cat". For the resources, available title, date, author and, publisher are saved in a text file with the name of the source folder followed by a "_misc" tag. Appendix C.3 exposes the structure of a sample directory.

Chapter 4

Learning Multi-Document News Summarization

Multi-Document summarization framework we developed consists of three parts: a sentence selector, a redundancy detector and an abstractive summarizer. In Subsection 2.2.2, we introduced the evolution of the Abstractive Multi-Document Summarization techniques. Even though there are some pure abstractive approaches, many of them still rely on extractive summarization to decide if the information is important enough to be included in the final summary. Hence, we assembled several techniques from extractive and abstractive summarization methods by considering their weakness, strength, and how they can exhibit better performance. Extractive approaches identify meaningful sentences, extract a subset of those based on compression ratio and apply little to no content modification. In abstractive approaches, systems employ natural language processing to provide a deeper semantic understanding of the text and write it from scratch (Desai and Rokade [2015]).

Each approach has its strengths and drawbacks in the case of summarizing news articles. We argue that using extraction methods to provide the salient information for the abstractive summarizer is an effective strategy and a unified summarizer can provide solutions for the challenges of both approaches. Thus in this chapter, we reveal the methods we used to make a unified multi-document summarizer for news articles alongside the evaluation techniques we employed to test the effectiveness.

Firstly, we introduce how we generated the training data for abstractive summarization by using extractive summarization methods. In Subsection 4.1.1, we introduce the extractive summarization method we used for this study, in Subsection 4.1.2 the strategies we devised to limit the input length is explained, in Subsection 4.1.3, we present the theory we employed to make a duplicated information filter and how we constructed the duplication filter, and in Subsection 4.1.4, the evaluation results for extractive summarization is detailed.

In Section 4.2, we present the abstractive summarization method we employed. We also explain how the method solves the problems of earlier approaches in neural abstractive summarization.

In Section 4.3, we explain the experimental setup and three different training methods that we devised: *double-abstractive, ea-full-target*, and *ea-short-target.* Finally, we present the evaluation results and provide several observations on the final summaries.

4.1 Extracting Salient Sentences

Reliability is one of the most important qualities when it comes to journalism. Since we decided to study a better method for summarizing news articles, we needed to be very careful about which information to keep and which to rule out. In this part of the study, we proposed an extractive summarization technique that we call the *Wikisummarizer*, which consists of two main components. The aim of the first component is scoring the sentences from the most important to the least. However, the scoring mechanism does not guarantee a duplication free output since the information is from multiple sources which may contain the same information that is presented in different ways. Therefore, in the second component, we added a mechanism to identify the relationship between selected sentences for filtering out the duplications.

In the following subsections, first, we introduce the summarization method we employed for extractive summarization. Next, we discuss the strategies we proposed to handle the input size. For this purpose, we devised two different strategies: *Diversity* and *Completeness.* Both strategies limit the input size to 100 sentences. Former one accepts the input in small chunks and favours the ones close to the beginning, whereas the letter strategy only accepts the input as a whole document. Later, we present the *cross-document relationship classification* (Radev and Zhang [2004]) and explain how we developed a filter to reduce the duplications based on the cross-document relationship classification. Finally, we reveal the evaluation results to expose the most suitable strategy to generate training data for abstractive summarization.

4.1.1 Sentence Scoring

As we mentioned in Chapter 3, Webis-wiki-corpus which we used for training, testing and evaluating the summarizer is merely retrieved from Wikinews and Wikipedia pages. From a broad range of different techniques and methods, we chose the extraction methods of a prior work that aims to create Wikipedia pages by summarizing the source documents.

A study by Google Brain (Liu et al. [2018]) proposes that generating English Wikipedia articles can be approached as a multi-document summarization of the documents initiated in the sources. It uses extractive summarization to find important information in the text, and in the following step it uses a neural abstractive model based on a decoder-only architecture to generate the Wikipedia article. This architecture is used due to its ability to scalably attend to sequences much longer than typical encoder-decoder architectures can. The study exhibited that using this approach, one can generate fluent, coherent multi-sentence paragraphs and even whole Wikipedia articles.

To provide an appropriate scoring mechanism for the Wikinews articles, we implemented the following ranking methods from the extractive component of the study (Liu et al. [2018]):

1. *TF-IDF*: This method is used to determine the strength of the relevance between the query words and the documents in a corpus (Ramos [2003]). In our study, we used the titles as query, the sentences as the documents, and the source article as the corpus to rank the sentences.

2. *TextRank* (Mihalcea and Tarau [2004]): An algorithm similar to PageRank (Brin and Page [1998]). Ranking is computed on a weighted graph where the nodes are text units and the edges are the similarity measure based on word overlap. We used sentences as text units.

3. *Sumbasic* (Nenkova and Vanderwende [2005]): The sentences are scored based on the frequency of the words they contain. After selecting the best scoring sentence, scores of the words in it are reduced. The algorithm iterates until the desired summary length is reached.

4. *Cheating*: A method which is implemented to improve the final quality of the summarization. The method ranks the sentences by using the recall of bigrams in the ground truth text with the following formula:

$$d(s_j^i, a_i) = \frac{bigrams(s_j^i) \bigcap bigrams(a_i)}{bigrams(a_i)} \tag{4.1}$$

   where s_j^i is the jth sentence of the ith article and a^i is the ith article.

After the system calculates the scores of each method mentioned above, it multiplies with each other to produce the final score for each sentence. Finally, it ranks them from the sentence with the highest score to the sentence with the lowest, and extracts first n sentences, where n is the desired summary length and fixed to 10 sentences in our study.

4.1.2 Strategies for Affordable Input Length: Diversity and Completeness

Since the combined length of the source documents are typically short enough, we did not encounter any problem during the first experiments. However, when we needed to summarize the entire document set to train the abstractive summarizer, some problems occurred due to computational limitations. Some articles in the document set have more than a hundred different sources, and some of them include a reasonable amount of sources, yet the source article consists of thousands of sentences. We provide the minimum, maximum and average amount of article and sentence in Table 4.1. In both cases, the dataset provides us with an extremely large amount of information which is most likely not useful and computationally almost impossible to process. Therefore, we decided to limit the input with a hundred sentences and decided upon testing two different strategies to extract the chunks which are most probably of higher importance.

The first strategy we developed is *completeness*. This strategy aims to include as many complete documents as possible. It starts with ordering the documents from the shortest to longest in terms of number of sentences to make sure that the process does not get stuck in the single-document scenario. Then it starts to include documents to the final document set until either it reaches the hundred-sentence limit or runs out of documents to include.

The second strategy (called *diversity*) considers the possibility that important information can be reserved in any of the documents in the document set. Thus, instead of including the complete documents, it includes small chunks of the documents in a specific order. We decided to set the size of the chunks five sentences long since it is long enough to form a paragraph of acceptable paragraph length[1]. Once the chunks are created, they are included to the final set starting from the first chunks of each document. When the system reaches the end of the document set for the first time, it proceeds with including the second chunks. At every iteration, it includes the next chunks in order until it includes all the chunks in the set or it reaches the hundred-sentence limit.

[1] `https://www.grammarly.com/blog/how-long-is-a-paragraph/`

Table 4.1: Minimum, maximum and average amount of resource per article and sentence per resource

Measure	Amount
Minimum resource assigned per article	1
Maximum resource assigned per article	196
Average number of resources assigned per article	2.17
Minimum amount of sentence an article contains	1
Maximum amount of sentence an article contains	8,827
Average amount of sentence an article contains	33.50

4.1.3 Removing Redundancy

In order to improve the quality of the final summary, we decided to exploit the Cross-Document Structure Theory (CST), which can identify the semantic relations between the sentences (Radev [2000]). In this subsection, firstly we introduce the CST enhanced summarization which inspired us to bring CST into our work, then CSTBank data that we used to train the system and lastly how we constructed the CST duplication filter.

When it comes to the multi-document summarization, the articles in the same cluster can exhibit some properties which need to be taken into account. Since the content of the articles is related, the sentences tend to be needless to appear in the final summary even though they get higher scores than others. A partial similarity can be observed (i.e. equivalence, subsumption, overlap or elaboration) as well as two sentences can be identical. Furthermore, since the information is of different news sources, it is also possible to find contradicting statements (Radev and Zhang [2004]).

We argue that an extractive summarization enhanced with CST duplication filter can provide a more robust method against repetitions due to the CST's capacity of identifying the similarity between sentences.

CST enhanced summarization

Radev and Zhang [2004] proposed a method to improve the quality of summaries by using auxiliary information from the text (i.e. lexical similarity) and cross-document relations. They stated the components of the CST Enhanced summarizer in their paper as follows (Radev and Zhang [2004]):

1. A scoring algorithm A_S that computes a numeric score for each sentence. Specifically, $score(S_i) = A_s(f_{1_i}, f_{1_i}, ..., f_{k_i})$, where f_1 through f_k are the features of each sentence.

2. A re-ranker R that adjusts sentence scores by looking at some other information (usually global, i.e., beyond the current sentence), such as lexical similarity or CST relationships between pair of sentences. Specifically, $score'(s_i) = score(s_i) + \Delta(S)$ where the adjustment is determined by certain global information with regard to S. Notice that Δ can be negative.

3. A compression ratio r, such that $0 \leq r \leq 1$.

4. A ranking algorithm A_R that selects the highest-score sentences, such that $N_{S'} = [N_S \cdot r]$ where N_S is the number of sentences in the original text and $N_{S'}$ is the number of sentences in the extract.

and they postulated the following hypothesis in their study (Radev and Zhang [2004]):

Hypothesis 1 Enhancing the CST connectivity of a summary will affect its quality as measured by relative utility.

Hypothesis 2 The effect of the enhancement will be dependent on the type of CST relationship added into the summary.

Their study was aiming to enhance the extractive summarization by selecting the sentences related to each other with CST relationships. However, our method is using the CST relationship analyzer they proposed and the dataset they used to detect the duplicated information.

Dataset for CST: CSTBank

CSTBank is a corpus of clustered documents each of which are manually annotated for CST relationships (Radev et al. [2003]). Some of the clusters in the corpora is acquired from secondary sources (i.e. DUC cluster is from the 2001 Document Understanding Conference training data) whereas some of them is compiled by the authors (i.e. The Milan 9 and Gulfair 11 clusters are retrieved from different news sites such as USA Today, MSNBC, CNN, FOX News, the BBC, the Washington Post and ABC News). Table 4.2 shows the statistics of the most up-to-date version of CSTBank which is published on the official website[2].

After the acquisition of the clusters, the experiment carried on to the manual annotation phase. For this task, eight judges had been hired to mark the CST relationship of eighteen types (Table 4.3). However, manual annotation is an expensive and difficult task with the full corpus. Not only that the large search space is hard to work with but also an agreement on the annotations is difficult to reach. For instance, the corpus used in the second step of the

[2] `https://yale-lily.github.io/downloads/clair/CSTBank/phase1.htm`

Table 4.2: Statistics of the CSTBank Phase I

Family	**Source(s)**	**Number of Clusters**	**Clustering method**	**Publicly available?**
duc01	DUC01 data	60	automatic	No
duc01trial	DUC01 sample data	4	automatic	No
duc02	DUC02 data	60	automatic	No
duc03	DUC03 data	60	automatic	No
hknews	HKNews corpus	40	automatic	No
manual	various online news agencies	10	manual	Yes
manual2	usenet groups	2	semi-manual	Yes
mds	online news agencies	6	manual	Yes
nie	NewsInEssence	50	automatic	No
novelty02	TREC2002 Novelty Track	53	automatic	No
other	misc.	1	automatic	No
tdt-pilot	Topic Detection and Tracking pilot data	25	automatic	No
tdt2	Topic Detection and Tracking 2	100	automatic	No

experiment contains 9 articles and 269 sentences which creates 18023 possible pairs to investigate. Hence, they decided to find the lexically similar sentence pairs to handle the problem. For this purpose, they experimented the following metrics and decided that World overlap is the most appropriate method (Radev and Zhang [2004]):

1. Word-based cosine similarity:

$$cos(s_1, s_2) = \frac{\sum s_{1,i} * s_{2,i}}{\sqrt{\sum (s_{1,i})^2} * \sqrt{\sum (s_{2,i})^2}} \tag{4.2}$$

2. Word overlap:

$$wol(s_1, s_2) = \frac{\#CommonWords(s_1, s_2)}{\#Words(s_1) + \#Words(s_2)} \tag{4.3}$$

3. Longest Common Subsequence:

$$lcs(s_1, s_2) = \frac{\#Words(LCS(s_1, s_2))}{\#Words(s_1) + \#Words(s_2)} \tag{4.4}$$

   where LCS is the number of words in the longest sequence that S_1 and S_2 have in common.

4. The BLEU metric: Linear combination of n-grams with length mismatch penalty(BP)

$$BP = \begin{cases} 1 : & \text{if } c > r \\ e^{1-\frac{r}{c}} : & \text{if} c \leq r \end{cases} \tag{4.5}$$

$$BLEU = BP * exp(\sum_{n=1}^{N} w_n log(p_n)) \tag{4.6}$$

where c is the length of the candidate sentence, r is the effective reference sentence length, p_n is n-grams precision, w_n is positive weights and N is the maximum n-gram length.

1,815 sentence pairs were produced and 1,145 CST-related pairs were identified out of the 18,023 sentence pairs mentioned in the last example. Later, judges have been given the CST annotation guidelines[3]. At least two judges had been assigned per cluster and allowed to assign multiple types of CST relationship per sentence to ensure the quality of annotations.

In our study, we used the *Milan9* and *Gulfair11* clusters of the *manual* corpora since those two have the biggest number of annotations in all of the clusters which are published online and manually generated. The number of the annotations for both clusters can be found in Table 4.4.

Table 4.3: Types of the cross-document relationships and examples (Radev and Zhang [2004])

ID	Relationship	Description	Text span 1 (S1)	Text span 2 (S2)
1	Identity	The same text appears in more than one location	Family members were asked to make identifications from photos taken after the bodies were recovered.	Family members were asked to make identifications from photos taken after the bodies were recovered.
2	Equivalence (Paraphrase)	Two text spans have the same information content	Six French government experts and an Airbus Industries representative flew in Thursday evening.	Six French government experts and a representative from Airbus Industries, the plane's manufacturer, flew in Thursday.
3	Translation	Same information content in different languages	Shouts of "Viva la revolucion!" echoed through the night.	The rebels could be heard shouting, "Long live the revolution".

Continued on next page

[3] https://yale-lily.github.io/downloads/clair/CSTBank/annotation_guide.pdf

Table 4.3 – *Continued from previous page*

ID	Relationship	Description	Text span 1 (S1)	Text span 2 (S2)
4	Subsumption	S1 contains all information in S2, plus additional information not in S2	Flight 072 crashed in shallow water near shore and Ali Ahmedi, a spokesman and an acting vice president for GulfAir, has said the pilot gave no indication to air traffic controllers that there were anyproblems in the plane.	He said there was no indication the pilot was anticipating an emergency landing.
5	Contradiction	S1 and S2 present conflicting information	The plane crashed into the 25th floor of the building.	The plane hit the 26th floor.
6	Historical Background	S1 gives historical context to information in S2	It was built in 1958 and designed by architects Gio Ponti and Pier Luigi Nervi.	The Pirelli Building in Milan, Italy was hit by a small plane.
7	Citation	S1 explicitly cites document S2	An earlier article quoted Prince Albert as saying "I never gamble."	Prince Albert then went on to say, "I never gamble."
8	Modality	S1 presents a qualified version of the information in S2, e.g., using "allegedly"	Sean "Puffy" Combs is reported to own several multi million dollar estates.	Puffy owns four multi million dollar homes in the New York area.
9	Attribution	S1 presents an attributed version of information inS2, e.g. using "According to CNN,"	A small airplace crashed into a government building in the heart of Milan, setting the top floors on fire, Italian police reported.	A small place crashed into the 25th floor of a skyscraper in downtown Milan today.
10	Summary	S1 summarizes S2	The Mets won the Title in seven games.	After a grueling first six games, the Mets came from behind tonight to take the Title.
11	Follow-up	S1 presents additional information which has happened since S2	But Pera's spokesman later said he had spoken with the Interior Minister and the crash didn't appear to be any kind of an attack.	In Rome, the senate's president, Marcello Pera,said it "very probably" appeared to be a terrorist attack.
12	Indirect speech	S1 indirectly quotes something which was directly quoted in S2	Mr. Cuban then gave the crowd his personal guarantee of free Chalupas.	"I'll personally guarantee free Chalupas," Mr.Cuban announced to the crowd.
13	Elaboration (Refinement)	S1 elaborates or provides details of some information given more generally in S2	50% of students are under 25; 20% are between 26and 30; the rest are over 30.	Most students at the University are under 30.
14	Fulfillment	S1 asserts the occurrence of an event predicted inS2	After traveling to Austria Thursday, Mr. Green re-turned home to New York.	Mr. Green will go to Austria Thursday.

Continued on next page

Table 4.3 – *Continued from previous page*

ID	Relationship	Description	Text span 1 (S1)	Text span 2 (S2)
15	Description	S1 describes an entity mentioned in S2	Police say the plane was an Air Commando, a small plane similar to a Piper.	A small plane crashed into a skyscraper in down-town Milan today, setting several floors of the 32-story building on fire.
16	Reader Profile	S1 and S2 provide similar information written for a different audience	The Durian, a fruit used in Asian cuisine, has a strong smell.	The dish is usually made with Durian.
17	Change of per-spective	The same entity presents a differing opinion or presents a fact in a different light	Giuliani criticized the Officer's Union as "too demanding" in contract talks.	Giuliani praised the Officer's Union, which provides legal aid and advice to members.
18	Overlap (partialequiv-alence)	S1 provides facts X and Y while S2 provides facts X and Z; X, Y, and Z should all be non-trivial.	The plane crashed into the 25th floor of the Pirelli building in downtown Milan.	A small tourist plane crashed into the tallest building in Milan.

Table 4.4: CST relationship statistics for the datasets used in this study

Relationship	Number of pairs in Milan9	Number of pairs in Gulfair11	Succesful Re-detection
Identity	194	0	95.87%
Equivalence (Paraphrase)	124	5	68.96%
Translation	0	0	-
Subsumption	312	47	46.30%
Contradiction	58	5	33.33%
Historical Background	101	163	10.09%
Citation	0	0	-
Modality	6	11	33.33%
Attribution	108	43	25.92%
Summary	5	6	16.66%
Follow-up	71	104	9.09%
Indirect speech	2	2	33.33%
Elaboration (Refinement)	5	333	10.96%
Fulfillment	1	9	0.00%
Description	221	83	17.21%
Reader Profile	0	0	-
Change of per-spective	0	0	-
Overlap (partialequivalence)	527	5	24.2%

Creating a duplication filter using CST

In this section, we specify how we developed the *CST duplication filter* based on Radev and Zhang [2004]'s automatic classification of CST relationships. The chosen algorithm to build a classifier is *boosting* due to its capacity of forming a *strong* classifier out of many *weak* classifiers which potentially can be obtained from the text. In the implementation, we decided to work with

ICSIBoost[4]. ICSIBoost is an open-source implementation of *Boostexter*, which is the method used in the original study (Radev and Zhang [2004]).

The features used as weak classifiers for boosting can be divided under three categories: lexical features, shallow syntactic-level features and deep syntactic-level features.

Lexical Features:

1. Number of tokens in sentence 1
2. Number of tokens in sentence 2
3. Number of tokens in common

A total of 3 weak classifiers are obtained from this category.

Shallow Syntactic-Level Features: For six different part of speech (POS) tokens (common noun, proper noun, verb, adjective, adverb, and cardinal number), the following metrics are found:

1. Number of POS_x tokens in sentence 1
2. Number of POS_x tokens in sentence 2
3. Number of POS_x tokens in common

where POS_x stands for the specific type of POS token stated before. A total of 18 weak classifiers are obtained from this category.

Deep Syntactic-Level Features: The only purpose of this category is making a heuristic approximation between sentences since an actual semantic analysis is still an ongoing AI-complete problem. To this end, the lexical-semantic distance between sentences are computed by the following steps:

1. Find the top level Noun Phrase (NP) and Verb Phrase (VP) for both sentences
2. Find the head tokens of NP and VP of both sentences
3. Align the heads
4. For each head pair (NP-heads & VP-heads) find the semantic distance described in Leacock and Chodorow [1998], J. Jiang and W. Conrath [1997], Resnik [1995], Lin et al. [1998] and Wu and Palmer [1994][5]

[4] `https://github.com/benob/icsiboost`

[5] In the original work Hirst and St-onge [1995] semantic distance is used, however, since the Stanford Parser library of Python does not support this package we decided to work with Wu and Palmer [1994]

A total of 10 weak classifiers are obtained from this category.

Milan9 and *Gulfair11* clusters provided us with the articles of the same topic alongside a CST judgement file called *sentrel* which contains the CST relationship annotations. Articles come in a specific format which marks the sentences with responding document id and a sentence id (See appendix D.4.1). This particular format allows us to read the annotations from a single file called *sentrel*. Sentrel files of the clusters contain the information of relationship types for each pair alongside the id of the judge, source document id (SDID), source sentence id (SSENT), target document id (TDID) and target sentence id (TSENT). The following XML snippet shows a sentence which is annotated with two different types of CST relationships[6] by two different judges.

```
<R SDID="36" SSENT="3" TDID="23" TSENT="16">
<RELATION TYPE="4" JUDGE="Z"/>
<RELATION TYPE="18" JUDGE="J"/>
</R>
```

In the original work there are two different scenarios for the classification (Radev and Zhang [2004]):

1. The binary classification: If the pair is annotated with CST relationship type assign "1" otherwise "0".

2. The full classification scenario: If the pair is annotated with CST relationship type assign CST relationship id otherwise "0".

Since we focused on developing a duplication filter, the goal of was detecting CST relationships rather than detecting the type of relationship, hence, we decided to work with the binary classification. We paired up every sentence in the cluster and processed each pair to produce a score set of the weak classifiers mentioned before. Then we assigned the information about the existence of CST relationship to the score set. The weak classification score set of a sentence pair is assigned with *cst* tag if the pair appears in the sentrel file and if it does not, then with *!cst* tag (See appendix D.1.1).

Next, we started to train ICSIBoost, which requires two different inputs for training. The first one is the *data* file that we produced out of the sentence pairs. The second is a *name* file which contains the name of classes and the name of features. After the training, it produces a *shyp* (strong hypothesis) file, which is used as a model for the classification. We provide the name file that we used for training in Appendix D.2.1 and a sample shyp file Appendix D.3.1.

[6] CST relationship type ids in the sentrel files are the enumeration of the full relationship list. Please see the Table 4.3 for the full list alongside the ids. In this example, TYPE="4" represents Subsumption and TYPE="18" represents Overlap

Tests and Improvements on Duplication Filter

Before we integrate the filter to the extractive summarizer we wanted to test it on the manually annotated articles to ensure that we use the best version. The first time we trained the system, we marked every sentence pair annotated in the sentrel file as CST-related. Then, we used the trained model on each sentence pair to examine if the system is able to detect the CST relationship, which was annotated manually. However, the results were not very promising and in need of further improvements. In Table 4.4, we provide the success rates of initial test alongside the CST types and manual annotation amounts in the datasets.

Next, we decided to keep only the CST types which we presumed to be the most useful for a duplication filter and with a promising success rate at the same time. To this end, we created individual CST relationship detectors for certain types: *equivalence, overlap, subsumption, description* and *elaboration*. For each type, we trained the system individually. For example, if the type is *overlap*, then the pairs of any type other than *overlap* are considered as not-CST-related whether they have a CST relationship or not. Individual training for the mentioned types exhibited significant improvement (Table 4.5), and those types are the only ones we used in the final version of the duplication filter.

Table 4.5: Success rates for the CST relationships, which are used in the final version of CST Duplication filter

Relationship	**Success Rate : First test**	**Success Rate : Final**
Equivalence	68.96%	79.31%
Subsumption	46.30%	84.56%
Elaboration	10.96%	46.45%
Description	17.21%	65.57%
Overlap	24.25%	61.97%

4.1.4 Evaluating the Sentence Extraction

As discussed earlier in this chapter, one of the goals of this study is making a unified summarizer for multi-documents. To this end, the Wikisummarizer manages to summarize a multi-document cluster into a single document summary. The summaries created by Wikisummarizer are then used to train the abstractive summarizer. Therefore, during the construction and evaluation, we considered the following aspects:

1. Being competitive against a baseline summarizer
2. Having an affordable extraction strategy for feeding the abstractive summarizer

As for the first aspect, we decided to use MEAD summarization platform (Radev et al. [2004]) as the baseline summarizer since it is one of the well known and publicly available multi-document summarization frameworks.

As explained in Section 4.1.2, when we tried to summarize the entire document set, we encountered some problems due to the size of some documents in the collection. To overcome this problem with the computational limits, we developed two strategies (Diversity and Completeness) which limit the size of the input down to a reasonable number of sentences. Therefore, once the Wikisummarizer proved itself competitive against the baseline algorithm, we tested those strategies and their versions upgraded with CST Duplication filter to find out which affordable extraction strategy has the best performance.

To evaluate the content produced by Wikisummarizer, we summarized the news articles from 145 clusters using all strategies and created the test sets of 145 summaries. Later, we applied ROUGE evaluation measure and made the comparisons to see if the Wikisummarizer meets the requirements of the aspects that we considered during the construction. Before looking into the results, let us mention what ROUGE is briefly.

ROUGE Evaluation Measure

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an automatic evaluation package for computer-generated summaries (Lin [2004]). ROUGE measure determines the quality of the summary by measuring the count of overlapping units in the text such as n-gram, word sequences, and word pairs between computer-generated summaries and the ground-truth summaries produced by humans.

As mentioned earlier, extractive summaries are used for training the abstractive summarizer. We argue that intuitively syntactic similarity has positive effects on semantic similarity; hence, closer the computer-generated summaries to their respective ground-truth summaries, the better training-set for the abstractive summarization is generated. To this end, we decided to use ROUGE-N, which measures by overlapping n-grams and ROUGE-L, which measures by the overlapping longest common subsequence. By doing so, we determine word-wise and sequence-wise similarity.

Experiments Against Baseline Summarizer

The goal of this experiment is to determine whether the performance of the Wikisummarizer is high enough to produce summaries to train the abstractive part. For this purpose, we generated a set of 145 summaries using both Wikisummarizer and the baseline algorithm. Figure 4.1 reveals that the precision and f-measure of the Wikisummarizer is higher than MEAD algorithm in almost every ROUGE type, whereas recall is slightly lower. Table 4.6 exposes the amount of improvement alongside the ROUGE scores. This result suggests that even though the Wikisummarizer is not a state-of-the-art extractive summarizer, it is still competitive. Therefore, for what accuracy in content generation concerns, it is a suitable candidate to produce summaries out of Wikinews articles for training an abstractive summarizer.

Table 4.6: ROUGE score comparsion for MEAD Sumarizer and Wikisummarizer

ROUGE TYPE	MEAD	Wikisummarizer	Improvement
ROUGE1-Recall	0.54	0.51	-0.03
ROUGE1-Precision	0.39	0.45	+0.06
ROUGE1-F Measure	0.40	0.47	+0.07
ROUGE2-Recall	0.19	0.18	-0.01
ROUGE2-Precision	0.14	0.16	+0.02
ROUGE2-F Measure	0.14	0.17	+0.03
ROUGE4-Recall	0.05	0.06	+0.01
ROUGE4-Precision	0.05	0.05	0.00
ROUGE4-F Measure	0.04	0.06	+0.02
ROUGEL-Recall	0.50	0.47	-0.03
ROUGEL-Precision	0.36	0.42	+0.06
ROUGEL-F Measure	0.37	0.44	+0.07

Determining the Best Strategy

This part of the experiment aims to find the most effective strategy amongst the extraction strategies (Diversity and Completeness) that we developed to cope with the source document size problem. To this end, we used the same set of 145 Wikinews clusters that we used for the previous stage (Wikisummarizer vs. MEAD) and generated summaries using each of our strategies and their CST duplication filter activated counterparts. Therefore, we had four different sets of summaries: diversity, completeness, CST duplication filter

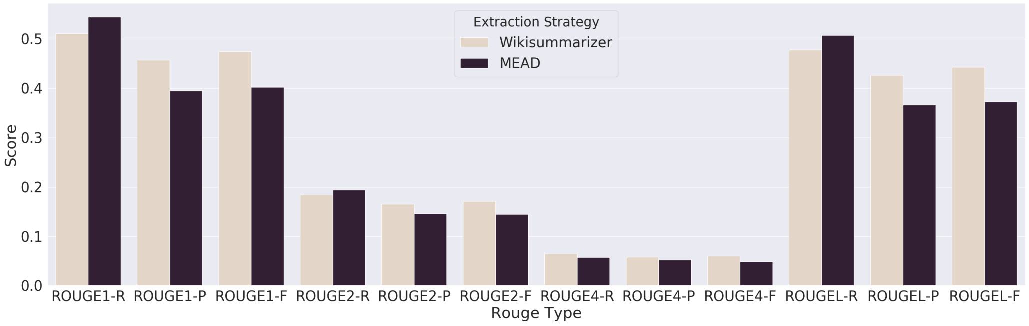

Figure 4.1: ROUGE score comparsion for MEAD Summarizer and Wikisummarizer

activated diversity (Div.+CST), and CST Duplication filter activated completeness (Compl.+CST). Table 4.7 shows the ROUGE scores and Figure 4.2 shows the comparison of each strategy.

As can be seen in Table 4.7, there is no significant difference between the extraction strategies. For each possible comparison, the difference between the scores is smaller than 0.05. Furthermore, there is no extraction strategy which gets the best scores for more than half of the ROUGE scores. These results show that no strategy is significantly more successful than others. Therefore, we determine the strategy based on different considerations.

First, we made a choice between diversity and completeness. The completeness provides the full content of the source document, whereas the diversity usually favours the chunks close to the beginning of the text. For the integrity of the information content, we decided to use the completeness strategy.

Next, we decided whether to use the CST duplication filter or not. We run the Wikisummarizer with completeness strategy on the Wikinews clusters for both the filter is activated and deactivated. Then we compared the summaries to detect the different results generated by CST duplication filter. We found out that 86.38% of the summaries are generated differently when the filter is on. Since the duplication filter changed the summaries for a significant amount of documents, we decided to use the filter.

We applied Wikisummarizer to 9,514 clusters that we acquired from the Wikinews. Unfortunately, our system faced with numerous errors during the extractive summarization such as characters causing a crash in tokenizer of *Python's NLTK* library or the sentences could not be parsed into POS tokens. We solved as many errors as possible; however, since it was impractical and extremely time-consuming to handle every single error in a document set of this size, we decided to use a *try-catch block* to prevent the crashes. As a result, we generated a set of 7937 extractive summaries to train the abstractive summarizer.

Table 4.7: ROUGE score comparsion for the extraction strategies: diversity, completenes, CST duplication filter activated diversity (Div.+CST), and CST duplication filter activated completeness (Comp.+CST)

ROUGE TYPE	Diversity	Completeness	Div.+CST	Comp.+CST
ROUGE1-Recall	**0.473**	0.463	0.434	0.438
ROUGE1-Precision	0.444	0.452	0.478	**0.483**
ROUGE1-F Measure	**0.424**	0.422	0.418	0.422
ROUGEL-Recall	**0.230**	0.226	0.215	0.213
ROUGEL-Precision	0.209	0.214	**0.231**	0.230
ROUGEL-F Measure	0.203	0.202	**0.204**	0.203

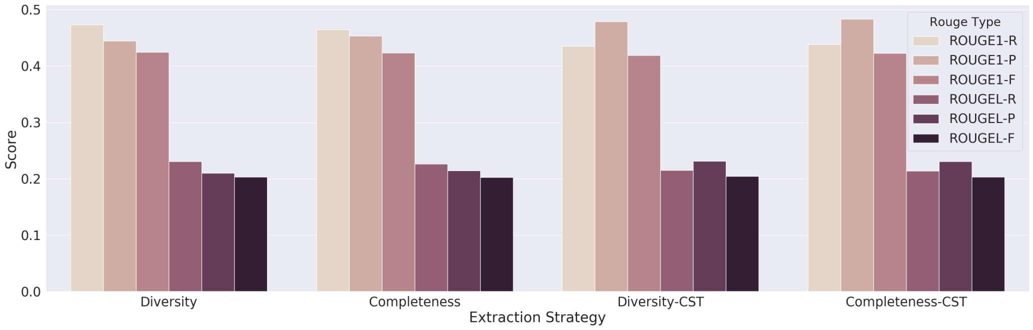

Figure 4.2: ROUGE score comparsion for different extraction strategies

4.2 Creating the Narrative

At the beginning of this chapter, we touched on the reason behind employing a unified approach for multi-document summarization task and in Section 4.1, we revealed how extractive summarization works and how it may help. However, even though the extractive summarization is useful to acquire relevant information, the summarizer still requires an effective abstractive approach to creating a good narrative. In this section, we introduce the neural abstractive summarization model that we employed alongside the solutions that the model offers for the problems of earlier approaches.

4.2.1 Neural Abstractive Summarization

See et al. [2017] proposed an enhanced sequence-to-sequence neural abstractive summarization method to cope with the problems such as repetitive words, inaccurate facts and senseless sentences. Their work did not just succeed to solve those problems of earlier approaches that mentioned in Subsection 2.1.3, but also showed improvement in ROUGE scores. Their model that we trained with the output of our extractive summarization method consists of three components: a baseline sequence-to-sequence model, a point-generator model to handle out-of-vocabulary words and coverage mechanism to eliminate repetition.

Sequence-to-sequence attentional model

Their baseline sequence-to-sequence model is similar to the model of Nallapati et al. [2016]. The model has an encoder which gets the tokens of the article (w_i) as input, and a decoder with a decoder state (s_t) which receives the word embeddings of the previous words at each time step. In this model, the *attention distribution* (a_t) helps the decoder to find the next word to generate and calculated as follows:

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \tag{4.7}$$

$$a^t = softmax(e^t) \tag{4.8}$$

where v, W_h, W_s and b_{attn} are learnable parameters. It is also used to produce the context vector (h_t^*) which can be considered as a fixed-size representation of the part read from the source at the current step.

$$h_t^* = \sum_i a_i^t h_i \tag{4.9}$$

Finally the vocabulary distribution (P_{vocab}) is calculated to predict the words as follows:

$$p_{vocab} = softmax(V'(V[s_t h_t^*] + b) + b') \tag{4.10}$$

where where V, V', b and b' are learnable parameters.

Pointer-Generator Network

Pointer-Generator Network is an extension of their baseline model which aims to handle the out-of-vocabulary words. In addition to the generating words from a fixed vocabulary, Pointer-Generator network can copy the words appearing in the source document via pointing. To do so, it calculates the generation probability (p_{gen}) for timestep t. Then, it uses p_{gen} to decide between generating or copying by the following probability distribution:

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \tag{4.11}$$

Finally, it calculates the general loss for the sequence with following function:

$$loss = \frac{1}{T} \sum_{t=0}^{T} (-log P(w_t^*)) \tag{4.12}$$

where w_t^* is the target word and t is timestep.

Pointer-Generator model (depicted in Figure 4.3) exhibited significant improvement over the baseline method. Summaries generated showed that the system is able to handle out-of-vocabulary words, copy the factual details with almost no mistake and produce a summary without fabricated facts.

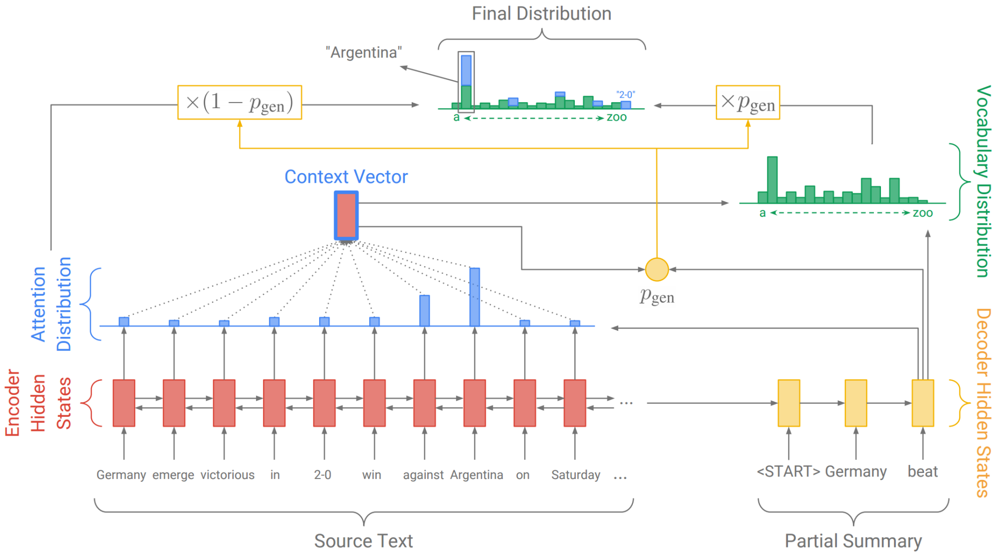

Figure 4.3: Pointer-Generator Network (See et al. [2017])

Coverage Mechanism

Coverage Mechanism is designed to prevent repetitions. In this model, *coverage vector* c^t is calculated to keep track of the words that have received attention so far:

$$c^t = \sum_{t'=0}^{t-1} a^{t'} \tag{4.13}$$

Next, it uses the coverage vector to inform the attention mechanism about its previous decisions by adding the coverage vector to the Equation 4.7, which is used to calculate attention distribution in the baseline model:

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + \mathbf{w_c c_i^t} + b_{attn}) \tag{4.14}$$

Finally, it penalizes the overlaps between coverage vector c^t and the new attention distribution a^t to prevent the repetitions:

$$covloss^t = \sum_i min(a_i^t, c_i^t) \tag{4.15}$$

4.3 Experiments and Evaluation

In Section 4.1, we explained how to make a multi-document extractive summarizer which aims to generate the training data from a clustered multi-document dataset. In this section, we investigate how effective using extractive summarization is to generate a training dataset for the Pointer-Generator Networks. We decided upon three different training models to conduct this experiment: *double-abstractive, ea-full-target* and *ea-short-target*. We trained the Point-Generator Network for each model until general loss and coverage loss become smaller than 0.1. For evaluating the final summaries, we focused on two aspects: content and readability. We generated 30 summaries per training model and applied the evaluation methods that we chose for both aspects.

4.3.1 Training Models

The first model is a trivial method to generate multi-document summaries with a method using only the Pointer-Generator Network. In this method that we call double-abstractive, we trained the system with the source articles and the ground-truth summaries as the target. Then we concatenated the summaries of the same cluster and used them as the training set with the same set of ground-truth summaries.

The second model employs our unified summarization (extractive + abstractive) approach. We trained the system with the extractive summaries as the input and the ground-truth summaries as the target. For this training method that we call ea-full-target, we set the length of the input summaries as 10 sentences and used the ground-truth summaries as a whole. This method aims to examine whether the extractive summarization can offer any improvement in multi-document summarization with the Pointer-Generator Network.

The third and last model that we call ea-short-target also uses the extractive summarization to generate input for the Pointer-Generator Network.

The only difference between this method and *ea-full-target* is the target size. We observed that the ground-truth summaries in the original work (See et al. [2017]) are much shorter than the input documents. However, they are almost the same size in most of the documents in our study. Thus, we wanted to see if a different ratio between the input and the target can make a difference. For this purpose, in this model, we used only the first three sentences of the ground-truth summaries for the target.

4.3.2 Automatic Evaluation for Content

As mentioned in Subsection 4.1.4, ROUGE determines the quality by the overlapping sequences. Although it has been criticized since it does not consider the expressed meaning and relies only on string matches, it is still the *de facto* evaluation system for computer-generated summaries (Lloret et al. [2018, p. 8]). We argue that the similarity between text-units can give an idea about the similarity between content. Therefore we decided to evaluate the content similarity by ROUGE measure. We provide the ROUGE scores of our training models in Table 4.8.

Table 4.8: ROUGE score comparsion for the training models

ROUGE TYPE	double-abstractive	ea-full-target	ea-short-target
ROUGE1-Recall	0.28	0.36	0.53
ROUGE1-Precision	0.20	0.26	0.58
ROUGE1-F Measure	0.23	0.29	0.54
ROUGEL-Recall	0.20	0.28	0.48
ROUGEL-Precision	0.15	0.19	0.53
ROUGEL-F Measure	0.16	0.21	0.49

Results showed that the unified summarizer trained with reduced ground-truth summaries achieved a significant improvement over the double-abstractive model, whereas the scores of its counterpart trained with the complete ground-truth summaries are barely higher than the double-abstractive. These results suggest the following:

1. A targeted multi-document abridgement method such as extractive summarization proved itself advantageous for generating content-wise more accurate summaries.
2. Proportionally shorter target summaries or more extended input summaries are more suitable for training pointer-generator frameworks.

4.3.3 Manual Evaluation for Readability

Even though the content-wise similarity is essential to asses the quality of a summary, it does not provide an overall consideration. Taking the readability into account is also important to understand how good the information is presented in a summary from the readers perspective (Lloret et al. [2018, p. 18]). Moreover, the work of Conroy and Dang [2008] showed that while ROUGE has a robust correlation with responsiveness for both model and computer-generated summaries, it is not able to account the significant gap in responsiveness between humans and systems.

Unlike the measurements for quality of the content, most of the work for determining the quality of readability focused on manual evaluation methods (Lloret et al. [2018, p. 23]). Hence we decided to evaluate the readability of the summaries manually, and for this purpose, we designed a survey (Survey 01) based on following criteria provided by Document Understanding Conference (DUC)[7](Pitler et al. [2010]):

Grammaticality: The summary should have no datelines,system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

Non-redundancy: There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice.

Referential clarity: It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.

Focus: The summary should have a focus; sentences should only contain information that is related to the rest of the summary.

Structure and Coherence: The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.

[7] `http://duc.nist.gov/`

Survey 01 consists of 3 groups of questionnaires. Every group has 3 questionnaires of 10 questions which ask the annotators to evaluate the summary for each criterion provided by DUC on a scale of five points: *1.Very poor, 2.Poor, 3.Barely acceptable, 4.Good, 5.Very good.* Each group contains the summaries of the same clusters in the same order. The only difference between the set of summaries for each group is the training method. For example, the first summary of *double-abstractive 01*,*ea-full-target 01* ,and *ea-short-target 01* are summaries of the documents from the same cluster produced by a summarizer trained by their respective method. We provide an overlook of the arrangement of the survey in Table 4.9.

Table 4.9: Arrangement of the Survey 01 for manual readability evaluation.

	Annotators								
Questionnaire	01	02	03	04	05	06	07	08	09
double-abstractive 01	x					x			
double-abstractive 02		x					x		
double-abstractive 03			x					x	
ea-full-target 01				x					x
ea-full-target 02	x				x				
ea-full-target 03		x				x			
ea-short-target 01			x				x		
ea-short-target 02				x				x	
ea-short-target 03					x				x

Before we calculate the mean scores for the summaries of each training model, we used DKPro Agreement[8] to test the reliability of the survey. DKPro Agreement is a software to define the level of agreement between the annotators, which supports various statistical methods to measure *inter-rater agreement* (Meyer et al. [2014]).

Amongst several different methods that DKPro Agreement provides, we decided to use *Cohen's Weighted κ* (Cohen [1968]) since it is listed as a *rater-specific* method in DKPro and a weighted method, therefore, supports the ordinal annotations. Unfortunately, inter-rater agreement results showed that there is little to none agreement between the annotators. Table 4.10 exposes that the Weighted κ scores for each aspect are very low, and in many cases, they are negative.

We suspected that the reason behind the incoherent answers is the annotators' lack of experience in linguistics. Thus, we designed another survey

[8] `https://dkpro.github.io/dkpro-statistics/`

Table 4.10: Inter-rater agreement scores of the Survey 01 for the manual readability aspects

Quest.	**Grammaticality**	**Non-redundancy**	**Referential clarity**	**Focus**	**Structure and Coherence**
01	-1.46	-0.27	-0.69	-0.57	-0.47
02	0.08	0.09	0.00	-0.10	0.07
03	0.49	0.32	0.19	0.12	-1.45
04	0.23	0.23	0.02	-0.09	-0.04
05	-0.48	0.20	0.27	0.23	0.11
06	-2.78	0.05	0.21	-0.23	-0.17
07	-0.08	-0.38	0.25	-0.25	-1.26
08	0.09	0.34	-0.17	0.03	-0.29
09	0.03	0.01	-0.17	-0.29	-0.13

Table 4.11: Arrangement of the Survey 02 for manual readability evaluation.

	Annotators								
Questionnaire	01	02	03	04	05	06	07	08	09
01	x			x			x		
02		x			x			x	
03			x			x			x

(Survey 02) which is easier to evaluate for those who have no expertise in linguistics and requires only fluency in English. In this new survey, we used the same set of 30 summaries for each training model, conjoined the summaries that are generated from the same cluster, and asked the annotators to rank them from the best to the worst. Table 4.11 exposes the arrangement of the new survey. We provided them with the ground-truth summaries to show how an ideal result should look like, and allowed them to consider the readability aspects from the previous survey. In order to align the results of both survey, we used the following ranking scheme:

The best summary: 5 points

The second best summary: 3 points

The worst summary: 1 point

According to several studies, inter-rater agreement score over 0.8 is considered as a reliable agreement (Landis and Koch [1977], Carletta [1996], Neuendorf [2002]). Inter-rater agreement scores in Table 4.12 shows that all questionnaires of the Survey 02 got a score higher than 0.8; hence, there is no conflict between annotators in the second survey that cannot be neglected.

Table 4.12: Inter-rater agreement between the questionnaires of Survey 02

Questionnaire	Weighted Cohen's κ
01	0.81
02	0.84
03	0.83

The mean manual evaluation scores in Table 4.13 shows that multi-document abstractive summarization using an extractive summarization method for abridgement outperforms the summarizer using only the abstractive method. This outcome is also consistent with the ROUGE scores of the training methods (Table 4.8). Nevertheless, we decided to measure the correlation between two different evaluation aspects to investigate how reliable the consistency is. For this purpose, we used *Pearson's correlation coefficient* (Hayter [c2002, p. 657]) to measure the strength of the linear correlation between ROUGE scores and manual evaluation results. As one can see in Table 4.14, most of the comparisons showed positive correlation for each training model.

Table 4.13: Mean manual evaluation scores for each training model

Training Model	Mean Score
double-abstractive	2.15
ea-full-target	2.67
ea-short-target	4.18

Table 4.14: ROUGE vs Manual Evaluation - Pearson Correlation scores for each training model.

ROUGE Measure	double-abstractive	ea-full-target	ea-short-target
Rouge-1 Regression	0.07	-0.01	0.31
Rouge-1 Precision	0.26	-0.04	0.39
Rouge-1 F Measure	0.22	0.09	0.36
Rouge-L Regression	0.08	0.01	0.35
Rouge-L Precision	0.23	-0.01	0.40
Rouge-L F Measure	0.21	0.11	0.38

Earlier, we proposed that an extractive summarization has a positive effect on content-wise accuracy. The positive correlation between the evaluation results for content similarity and readability suggests that providing a content-wise similar input text with ground-truth summaries is not only desirable for producing content-wise accurate abstractive summaries but also helps with better readability. The scatterplot matrix in Figure 4.4 reveals the comparison of the evaluation results for each measurement. Data points in each scatterplot represent two different evaluation score for every question in Survey 02. Furthermore, data points and correlation lines are color-coded for each training methods. It is possible to see that regardless of the training model, summaries with more similar content usually is ranked as more readable by the human annotators.

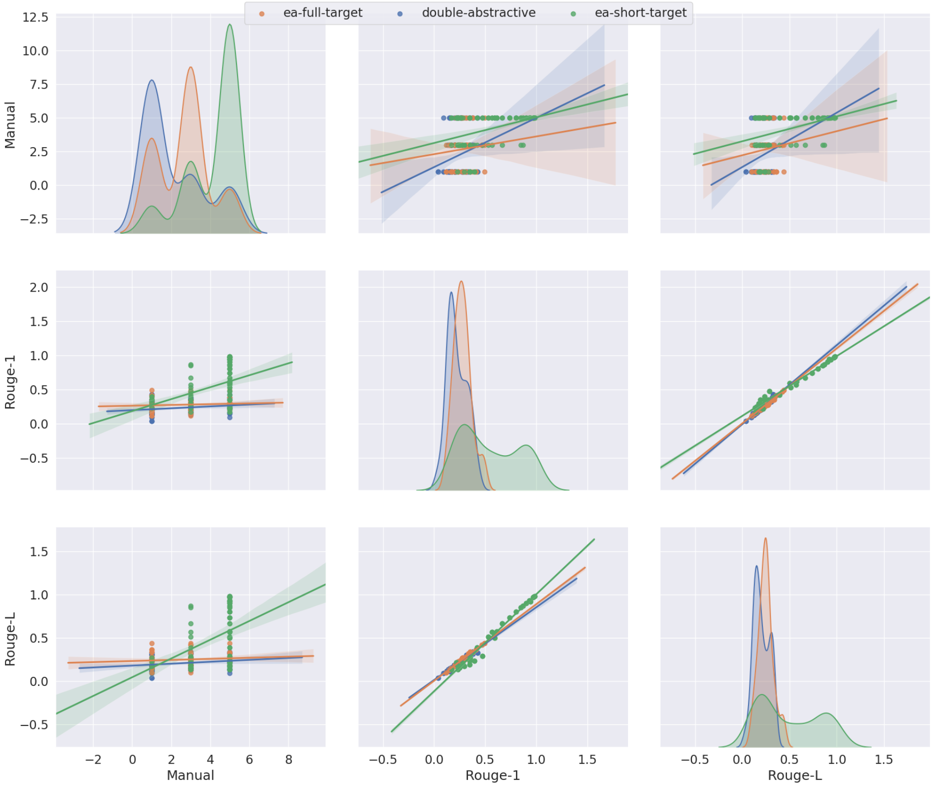

Figure 4.4: Comparsion of ROUGE-1, ROUGE-L and Manual evaluation scores for all training methods

4.3.4 Observations on the Final Summaries

In the previous subsection, we showed that training the abstractive summarizer with the dataset created by multi-document extractive summarization outperforms the model using only the Pointer-Generator Network for multi-document abstractive summarization. However, the unified summarization method we proposed is still far from being ideal. Hence, we want to provide several observations on the summaries generated by our best model (ea-short-target) to exhibit its advantages and drawbacks.

As mentioned earlier, the pointer-generator model that we used for abstractive summarization introduced solutions for repetitive words, inaccurate facts and senseless sentence. Unfortunately, numerous summaries generated in our study contains these errors.

The example shown in Table 4.15 exhibits the failure in finding out-of-vocabulary words. In this case, the system is failing to generate the uncommon or unique tokens such as **lf, 435-foot, bacsick** and **all-time career** and replacing them with unknown tag ([UNK]).

Table 4.15: Example of [UNK] error

Input: San Francisco Giants Barry Bonds speaks to the crowd as former Giants great Willie Mays stands by his side.Photo: ReutersBarry Bonds hit the 756th home run of his career to set a new Major League record on Tuesday, sparking wild celebrations among his hometown fans and mixed reaction elsewhere because of past steroid allegations.On Wednesday, as some tried to rain on the celebration with talk of Bonds and steroids, Easton was in no mood for it.In belting his landmark homer against the Washington Nationals, the 43-year-old San Francisco Giants slugger eclipsed the mark set by Hank Aaron in 1974."I don't pay attention to it," Easton says.Bonds, in his 22nd Major League season, completed the feat off Nationals starting pitcher Mike Bacsik in the fifth inning to put the Giants 5-4 ahead in a game they would go on to lose 8-6."Let the courts sort it out and let history sort it out.In a recorded message broadcast on the stadium's video board, Aaron paid tribute to Bonds.Right now, I just want to live for the moment like he does."Near the corner of Third and King in San Francisco at AT&T; Park, about a hundred people are waiting in line at the team souvenir shop — all of them looking to buy memorabilia of Bonds' historic home run."Throughout the past century the home run has held a special place in baseball and I have been privileged to hold this record for 33 of those years," he said.
Ground-truth Summary: yesterday san francisco giants lf barry bonds hit a 435-foot home run , his 756th , off a pitch from mike bacsick of the washington nationals , breaking the all-time career home run record , formerly held by hank aaron.the pitch , the seventh of the at-bat , was a 3-2 pitch , which bonds hit into the right-center field bleachers.matt murphy , a 22-year-old from queens in new york city , got the ball and was promptly protected and escorted away from the mayhem by a group of san francisco police officers .
Computer-Generated Summary: yesterday san francisco giants [UNK] barry bonds hit a [UNK] home run , his 756th , off a pitch from mike [UNK] of the washington nationals , breaking the [UNK] home run in 1974 .

The computer-generated summary in Table 4.16 is another case that the system fails to produce the correct words. In this case, the system replaces the word **chief** with a more common word **police**. Moreover, it copies the part "in 1974" from the input incorrectly. Both problems cause the generation of inaccurate information. Considering the goal of this study is summarizing news, this issue must be taken care of in further research.

Table 4.16: Example of an inaccurate fact in a summary

Input: The armband is large, bright pink and has a Hello Kitty motif with two hearts embroidered on it.From today, officers who are late, park in the wrong place or commit other minor transgressions will have to wear it for several days.The armband is designed to shame the wearer, police officials said."This is to help build discipline.We should not let small offences go unnoticed," Police Colonel Pongpat Chayapan told Reuters news agency."Guilty officers will be made to wear the armbands in the office for a few days, with instructions not to disclose their offences.Let people guess what they have done," he said.Further offences would be dealt with using a more traditional disciplinary panel, he said.The cartoon character Hello Kitty was first introduced by Japanese company Sanrio **in 1974**.The cute round-faced cat has become an Asia-wide marketing phenomenon, with Hello Kitty products such as stationery, hair accessories and kitchen appliances available across the region.
Ground-truth summary: the armband , which features "hello kitty" sitting on top of two hearts , will be worn by police officers who commit minor offences.these include , and parking in a prohibited area.the officers will also be forced to stay with **the deputy chief all day in division office** and will be forbidden to disclose their offences.
Computer-generated summary: the armband , which features "hello kitty" sitting on top of two hearts , will be worn by police officers who commit minor [UNK] include , and parking in a prohibited area.the officers will also be forced to stay with **the deputy police in 1974**.

Another shortcoming of the system is repetition. At the end of the summary in Table 4.17, the system is repeating the same phrase. The repeated text *"seas and i order you to serve five years in prison"* is a part of the input and the ground-truth summary; therefore it is related to the content of the final summary. However; since the coverage mechanism probably failed to see the words already received attention from the attention mechanism, the system does not know that the words are already covered and keeps generating the cycle of same words.

Nevertheless, unsuccessful results are not the only case. As a matter of fact, the system succeeded to produce an accurate and correct summary in numerous cases. One can observe in Table 4.18 that the generated summary is precisely the same text with the ground-truth summary, and in Table 4.19, the generated summary is almost the same except for the correct paraphrasing in the last sentence. These successful summaries prove that the system is capable of providing accurate and well-written summaries.

The Pointer-Generator Network in the original work (See et al. [2017]) is trained with CNN/Daily Mail data set. Considering this data set has approximately 287k documents[9] whereas our training data set has only 7,937 documents, it is safe to argue that the number of the successful summaries can be higher with the expanded version of the dataset we created.

[9] https://cs.nyu.edu/~kcho/DMQA/

Table 4.17: Example of a repetition in computer-generated summaries

Input: A Kenyan court has sentenced seven Somali pirates to five years in prison for attacking a Spanish ship last year.The seven are the third pirate gang jailed since foreign warships started patrolling the area two years ago.They have been held in Mombasa since being captured by the Spanish navy while trying to hijack the Maltese-flagged merchant ship Anny Petrakis.In recent years, pirates have extended their reach further from the shores of East Africa.Although the pirates' success rate has fallen due to patrols, attacks have continued, and suspects are regularly freed because of doubts about where they should face trial."I have concrete proof that you attacked a vessel in the high **seas and I order you to serve five years in prison**," the presiding magistrate, Timothy Ole Tanchut, said.Mr Tanchut said the men would be sent back to Somalia after serving their terms.About 15 pirates are serving prison sentences in Kenya and some 100 are in custody.Kenya has signed agreements with the European Union relating to the prosecution of piracy suspects, but has said that other countries must agree to share the "burden".The court in Mombasa was funded by international donors, including the United Nations Office on Drugs and Crime, the European Union, Australia and Canada.
Ground-truth summary: A court in , Kenya has sentenced a group of seven Somali pirates to five years each in jail, according to a statement by the European Union.Anti-piracy mission EU Navfor said the were arrested by Spanish navy sailors after attacking Malta-registered cargo vessel "Anny Petrakis".Presiding over the conviction and sentencing, Timothy Ole Tanchut told the men he "...[had] concrete proof that you attacked a vessel in the high **seas and I order you to serve five years in prison**," ruling that they will be deported to Somalia after serving their sentences.
Computer-generated summary: a court in , kenya has sentenced a group of seven somali pirates to five years each in jail , according to a statement by the european [UNK] mission eu [UNK] said that the men , " i have concrete proof that you attacked a vessel in the high **seas and i order you to serve five years in prison seas and i order you to serve five years in prison seas and i order you to serve five years in prison seas and i order you to serve five years in prison seas and i order you to serve**

Table 4.18: Example of the 100% accurate summary

Input: BEIJING (Reuters) - At least 28 people were killed by knife-wielding attackers in a "violent terrorist attack" at a train station in the southwestern Chinese city of Kunming, and police shot dead five of the assailants, state media said on Sunday.Another 113 people were wounded, the official Xinhua news agency said, revising down a previous higher figure.It said the attack had taken place late on Saturday evening."It was an organized, premeditated violent terrorist attack," Xinhua said.Police were searching for around five others of the unidentified attackers, it said.Kunming resident Yang Haifei told Xinhua that he was buying a ticket when he saw a group of people, mostly wearing black, rush into the station and start attacking bystanders."I saw a person come straight at me with a long knife and I ran away with everyone," he said, adding that the attackers caught those who were slower."They just fell on the ground." Graphic pictures on the Twitter-like microblogging service Sina Weibo showed bodies covered in blood lying on the ground at the station.State television showed police wrapping a long, sword-like knife in a plastic bag, amid heavy security at the station.There was no immediate word on who was responsible.
Ground-truth summary: At least ten attackers with knives, dressed in black, attacked a train station in , China yesterday.At least 28 victims were killed, with 113 more wounded by knives, Chinese state news agency reported.The local municipal government accuses " separatist forces" for the attack.
Computer-generated summary: At least ten attackers with knives, dressed in black, attacked a train station in , China yesterday.At least 28 victims were killed, with 113 more wounded by knives, Chinese state news agency reported.The local municipal government accuses " separatist forces" for the attack.

Table 4.19: Example of a summary which paraphrases correctly

Input: Operatives had only a 10-second window to hit the satellite - USA 193 - which went out of control shortly after it was launched in December 2006.4 days agoUS Defence Secretary Robert Gates has said the United States is prepared to share with China some of the information it has about its missile strike on an ailing satellite.ARLINGTON, VA – At 10:29 p.m. last evening the Navy confirmed it hit a falling and potentially dangerous defense intelligence satellite using an SM-3 missile fired from the deck of the USS Lake Erie in the Northern Pacific.WASHINGTON — A missile interceptor launched from a Navy warship has struck a dying American spy satellite orbiting 130 miles over the Pacific Ocean, the Pentagon announced late Wednesday.His comments came after Beijing complained the missile strike could cause harm to outer space security and some countries.Officials were worried its hydrazine fuel could do harm, but it is not yet known if the fuel tank was destroyed.Officials cautioned that while early information indicated that the interceptor's "kill vehicle" had hit the satellite, it would be 24 hours before it could be determined whether the fuel tank with 1,000 pounds of toxic hydrazine had been destroyed as planned.Officials say the missile likely destroyed its intended target, a 1,000 pound tank of toxic hydrazine fuel.Mr Gates told reporters during a visit to the state of Hawaii that the US is prepared to share whatever it can "appropriately" share with China.Even so, one official who received a late-night briefing on the mission expressed confidence that the impact had been so powerful that the fuel tank probably had been ruptured.
Ground-truth summary: the united states navy has successfully destroyed a crippled spy satellite in a decaying orbit , by intercepting it with a missile.a modified sm-3 missile was launched from the uss lake erie at 03:26 gmt this morning , and intercepted the usa-193 satellite around three minutes later.it has been reported that **the satellite has broken into around 80 pieces** , some of which have already re-entered the earth 's atmosphere .
Computer-generated summary: the united states navy has successfully destroyed a crippled spy satellite in a decaying orbit , by intercepting it with a missile.a modified sm-3 missile was launched from the uss lake erie at 03:26 gmt this morning , and intercepted the usa-193 satellite around three minutes later.it has been reported that **the satellite has been damaged**.

Chapter 5

Conclusion

In this study, we investigated the methods for making a multi-document summarizer for news articles. The first contribution of the thesis is the *Webis-wikinews-corpus*. We created the *Webis-wikinews-corpus* out of 9,514 Wikinews /21,314 source articles and 2,174 Wikipedia /17,807 source articles to provide a corpus consisting of clustered news articles with their respective human-generated summaries. We believe the corpus can be used for training clustering or classification frameworks for news corpora as well as it is utilized for multi-document summarization in this study.

The next contribution of this work is a multi-document extractive summarizer called *Wikisummarizer*. For the first component of the Wikisummarizer, we reproduced the extractive summarizer of another work (Liu et al. [2018]) which is used for generating Wikipedia articles out of sources on Wikipedia pages. The second component of the Wikisummarizer is a filter which excludes the duplicated information by finding the semantic relations between sentences based on *cross-document structure theory*. We also proposed two different input size limitation strategies to deal with the size of the source documents. Although we proposed a multi-document extractive summarization method for general usage, in this work, we used it to produce the training data for abstractive summarization.

The last contribution is the evaluations and observations on the quality of our unified approach. Our experimental setup consists of three different training models, one of which uses only abstractive summarization and the other two are based on the unified method as we intended to make. The only difference between the unified method based models is the size of the target document. The purpose of the first model is examining whether a targeted abridgement method such as extractive summarization achieves an improvement over using only abstractive summarization. Moreover, we proposed the third method with a shorter target size to examine if a greater ratio between input and the target can achieve to produce better summaries. On the final summaries, we conducted an investigation that consists of the following elements:

1. Automatic Evaluation with ROUGE measure to evaluate the content-wise accuracy
2. Manual Evaluation to evaluate the readability aspect
3. Observations on the final summaries to find out how successful the proposed summarizer using our corpus is.

Evaluation results showed that a targeted abridgement method outperforms the method using only the Pointer-Generator Networks both in content and readability aspect. Moreover, the positive correlation between the results of ROUGE and the readability survey implied that a better content selection for the training set results in generating more readable abstractive summaries. Next, we compared the results of unified summarizers with different target sizes. The one with shorter ground-truth summaries showed a significant improvement over the one with full-size ground-truth summaries. These results exhibited that the greater the ratio between input and the target, the better results are produced. Finally, we made some observations on the final summaries of our best method. We noticed that the final summaries are still producing the errors that Pointer-Generator Network model promises to solve. Despite some erroneous results, the framework is still capable of generating proper summaries, which means there is still room for improvement, yet the idea behind is promising. Therefore, we propose several tasks for future work.

As mentioned in section 4.3.4, the dataset which is used to train Pointer-Generator Network model has approximately 287k documents, whereas our training data set has only 7,937 documents. We argue that the difference between the dataset sizes is one of the possible reasons behind the erroneous results. Testing our unified summarization approach with a bigger dataset to interrogate this assumption can be considered for future work.

Another future work we want to propose is about Webis-wikinews-corpus. The corpus we created provides a document set of clustered news. Using the corpus as training data, one can train a clustering or classification framework. Furthermore, the corpora can be expanded by using the trained clustering/ classification model.

Another improvement on the corpus can be made by using Wikipedia sources. We decided to exclude the Wikipedia documents since the parts which are the interest of this project are scattered through very long documents. A strategy can be devised to find and gather the chunks related to the news sources, therefore expand the size of our Webis-wikinews-corpus.

In their recent work, Fabbri et al. [2019] pointed out the shortage of extensive multi-document summarization datasets, and introduced the first large-scale multi-document dataset in news domain. The dataset they called Multi-News consists of 56,216 clusters and over 250,000 source article links. Each cluster contains the human-written summaries from *newser.com* and the Wayback-archive links of the cited documents. Due to the similarity of the content and domain between Webis-wikinews-corpus and Multi-News, we argue that two datasets can be merged to be used in future work.

Lastly, we suggest using different options of the Wikisummarizer to explore the effects on the abstractive summarization. We developed two different input size limitation algorithms and integrated a duplication filter. The main focus of the study is the unified summarizer but not the duplication filter or the limitation algorithms. Thus we chose the strategy which we deemed to be the most appropriate and used it to generate the training data. However, the effectiveness of training data generated by other possible options is still an open research question. The pointer-generator network model can be tested with the training data created by other permutations of the algorithms and the filter (i.e. diversity with CST duplication filter, completeness without CST Duplication Filter etc.)

Appendix A

Sample XML entry for the news page from English Wikinews Dump file 2017

```
<page>
    <title>President of China lunches with Brazilian President</title>
    <ns>0</ns>
    <id>736</id>
    <restrictions>move=sysop:edit=sysop</restrictions>
    <revision>
      <id>4198935</id>
      <parentid>4198883</parentid>
      <timestamp>2016-03-05T13:06:11Z</timestamp>
      <contributor>
        <username>Pi zero</username>
        <id>39076</id>
      </contributor>
      <minor />
      <comment>added [[Category:Petrobras]] using [[Help:Gadget-HotCat|HotCat]]</comment>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">{{date|November 13, 2004}}
            {{Brazil}}[[w:Hu Jintao|Hu Jintao]], the [[w:President of the People's Republic of China|President]] of the         [[People's Republic of China]] had lunch today with the [[w:President of Brazil|President]] of [[Brazil]],           [[w:Luiz InÃ¡cio Lula da Silva|Luiz InÃ¡cio Lula da Silva]], at the ''Granja do Torto'', the President's country residence in the [[w:Brazilian Federal District|
   Brazilian Federal District]]. Lunch was a
   traditional Brazilian           [[w:barbecue|
   barbecue]] with different kinds of meat.

Some Brazilian ministers were present at the event:
    [[w:Antonio Palocci|Antonio Palocci]] (Economy)
   ,          [[w:pt:Eduardo Campos|Eduardo Campos
   ]] ([[w:Ministry of Science and Technology (
   Brazil)|Science and Technology]]),          [[
   w:JoÃ£o Roberto Rodrigues|Roberto Rodrigues]] (
   Agriculture), [[w:pt:Luiz Fernando Furlan|Luiz
   Fernando Furlan]]          (Development), [[
   w:Celso Amorim|Celso Amorim]] ([[w:Ministry of
   External Relations (Brazil)|Exterior Relations
   ]]),          [[w:Dilma Rousseff|Dilma Rousseff
   ]] (Mines and Energy). Also present were [[
   w:pt:Roger Agnelli|Roger Agnelli]]         ([[
   w:Vale (mining company)|Vale do Rio Doce]]
   company president) and Eduardo Dutra ([[
   w:Petrobras|Petrobras]],         government oil
   company, president).

This meeting is part of a new [[w:political economy
   |political economy]] agreement between Brazil
   and China where           Brazil has recognized
   mainland China's [[w:socialist market economy|
   market economy]] status, and China has
             promised to buy more [[w:economy of
   Brazil|Brazilian products]].

{{haveyoursay}}
== Sources ==
{{wikipedia|Workers' Party (Brazil)|Brazilian
   Workers's Party}}
*{{source|url=http://br.news.yahoo.com/041113/25/
   p0en.html
 |title=
 |author=
 |pub=Agencia Estado
 |date=November 13, 2004}} {{Source offline}} {{pt
    }}
*{{source|url=http://www.highbeam.com/doc/1P1
   -102429439.html
 |title=Chinese president treated to typical
    Brazilian barbecue
 |author=Associated Press
 |pub=HighBeam Research
 |date=November 13, 2004}}
*{{source|url=http://news.bbc.co.uk/2/low/americas
```

```
             /4008499.stm
           |title=Brazil backs China on trade bid
           |author=
           |pub=BBC News
           |date=November 12, 2004}}
          *{{source|url=http://www.chinadaily.com.cn/english/
             doc/2004-05/24/content_333379.htm
           |title=Brazil sees market economy in China
           |author=
           |pub=China Daily
           |date=May 24, 2004}}

          == External links ==
          * [http://www.brasil.gov.br/ Brazilian government
             website] {{pt}}
          * [http://www.embchina.org.br/por/ China embassy in
              Brazil (in Portuguese and Chinese)]

          {{publish}}
          {{archived}}
          {{PD-Article}}

          [[Category:Brazil]]
          [[Category:China]]
          [[Category:India]]
          [[Category:Politics and conflicts]]
          [[Category:Hu Jintao]]
          [[Category:Luiz InÃ¡cio Lula da Silva]]
          [[Category:South America]]
          [[Category:Asia]]
          [[Category:Dilma Rousseff]]
          [[Category:Petrobras]]</text>
                <sha1>2d0obiwiaixjp5aqohoset5d4dcmgla</sha1>
  </revision>
</page>
```

Appendix B

Code snippet to generate JSON and XML output

```
def write_output(self, out, text):
        url = get_url(self.id)
        if options.write_json:
            json_data = {
                'id': self.id,
                'url': url,
                'title': self.title,
                'text': "\n".join(text),
                'sources': self.sources,
                'externals': self.externals,
                'categories': self.categories,
                'reporters': self.reporters
            }
            if options.print_revision:
                json_data['revid'] = self.revid
            out_str = json.dumps(json_data, ensure_ascii=False)
            if out == sys.stdout:   # option -a or -o -
                out_str = out_str.encode('utf-8')
            out.write(out_str)
            out.write('\n')
        else:
            if options.print_revision:
                header = '<doc id="%s" revid="%s" url="%s"
                   title="%s">\n' % (self.id, self.revid, url,
                   self.title)
            else:
                header = '<doc id="%s" url="%s" title="%s">\n'
                   % (self.id, url, self.title)
            footer = "\n</doc>\n"
            if out == sys.stdout:   # option -a or -o -
                header = header.encode('utf-8')
            out.write(header)
            out.write("<text>\n")
            for line in text:
                if out == sys.stdout:   # option -a or -o -
                    line = line.encode('utf-8')
                out.write(line)
                out.write('\n')
            out.write("</text>\n")
            for source in self.sources:
                out.write('<source>%s</source>\n' % (source))
            for externallink in self.externals:
                out.write('<externallink>%s</externallink>\n' %
                    (externallink))
            for category in self.categories:
                out.write('<category>%s</category>\n' % (
                   category))
            for reporter in self.reporters:
                out.write('<reporter>%s</reporter>\n' % (
                   reporter))
            out.write(footer)
```

Appendix C

Miscellaneous Information

C.1 Namespaces

Table C.1: Namespaces of Wikinews Dump File

Namespace Id	Namespace
-2	Media
-1	Special
0	-
1	Talk
2	User
3	User talk
4	Wikinews
5	Wikinews talk
6	File
7	File talk
8	MediaWiki
9	MediaWiki talk
10	Template
11	Template talk
12	Help
13	Help talk
14	Category
15	Category talk
90	Thread
91	Thread talk
92	Summary
93	Summary talk

Continued on next page

Table C.1 – *Continued from previous page*

Namespace Id	Namespace
100	Portal
101	Portal talk
102	Comments
103	Comments talk
446	Education Program
447	Education Program talk
828	Module
829	Module talk
2300	Gadget
2301	Gadget talk
2302	Gadget definition
2303	Gadget definition talk
2600	Topic

C.2 Full list of the News Websites from allsides data collect

Table C.2: Full list of the News Websites from allsides data collect

Column 1	Column 2
vanityfair.com	bloomberg.com
vice.com	sfgate.com
democracynow.org	spectator.org
mediaite.com	dailykos.com
sunlightfoundation.com	reason.com
politifact.com	mrc.org
opensecrets.org	watchdog.org
techcrunch.com	msnbc.com
independent.co.uk	cato.org
bostonglobe.com	thenation.com
pando.com	nationalinterest.org
time.com	thedailybeast.com
hotair.com	ksl.com
mashable.com	ijr.com
chicagotribune.com	usnews.com
theweek.com	nbcnews.com
cbn.com	cbsnews.com

Continued on next page

Table C.2 – *Continued from previous page*

Column 1	**Column 2**
newsweek.com	factcheck.org
theatlantic.com	abcnews.go.com
abc.net.au	bbc.com
bbc.co.uk	csmonitor.com
motherjones.com	mediamatters.org
jeffjacoby.com	nymag.com
lasvegassun.com	theblaze.com
nationalreview.com	vox.com
usatoday.com	washingtonpost.com
reuters.com	latino.foxnews.com
foxbusiness.com	foxnews.com
businessinsider.com	huffingtonpost.com
people-press.org	cnsnews.com
nypost.com	nytimes.com
washingtontimes.com	realclearpolitics.com
newsmax.com	thehill.com
thinkprogress.org	cnn.com
washingtonexaminer.com	pbs.org
theguardian.com	breitbart.com
slate.com	npr.org
salon.com	townhall.com
wnd.com	politico.com
dailywire.com	buzzfeed.com
latimes.com	forbes.com
nasa.gov	

C.3 Directory Structure

```
wn_5140
├── wn_5140.txt
├── wn_5140_cat.txt
├── wn_res_5140_00
│   ├── wn_res_5140_00.txt
│   └── wn_res_5140_00
│       └── wn_res_5140_00_misc.txt
└── wn_res_5140_01
    ├── wn_res_5140_01.txt
    └── wn_res_5140_01
        └── wn_res_5140_01_misc.txt
```

Appendix D

CST Duplication Filter

D.1 ICSIBOOST sample data file

D.1.1 A snippet from milan9-gulfai-binary.data

```
...
11 ,11 ,1 ,4 ,3 ,0 ,0 ,1 ,0 ,2 ,2 ,0 ,0 ,0 ,0 ,0 ,2 ,0 ,0 ,0 ,0
     ,0.864997437487 ,0.860201265223 ,0.0548572695074
     ,0.0515151927238 ,-0.0 ,0 ,-0.0 ,0.0 ,0.117647058824
     ,0.166666666667 , !cst
11 ,11 ,1 ,4 ,3 ,0 ,0 ,1 ,0 ,2 ,2 ,0 ,0 ,0 ,0 ,0 ,2 ,0 ,0 ,0 ,0
     ,0.864997437487 ,0.860201265223 ,0.0548572695074
     ,0.0515151927238 ,-0.0 ,0 ,-0.0 ,0.0 ,0.117647058824
     ,0.166666666667 , !cst
11 ,11 ,1 ,4 ,3 ,0 ,0 ,1 ,0 ,2 ,2 ,0 ,0 ,0 ,0 ,0 ,2 ,0 ,0 ,0 ,0
     ,0.864997437487 ,0.860201265223 ,0.0548572695074
     ,0.0515151927238 ,-0.0 ,0 ,-0.0 ,0.0 ,0.117647058824
     ,0.166666666667 , !cst
11 ,11 ,11 ,4 ,4 ,4 ,2 ,2 ,2 ,1 ,1 ,1 ,1 ,1 ,1 ,0 ,0 ,0 ,0 ,0 ,0
     ,3.63758615973 ,3.25809653802 ,1e+300 ,1e+300 ,12.2683753357
     ,6.78175283693 ,1.0 ,1.0 ,1.0 ,1.0 , cst
...
```

D.2 ICSIBOOST names file

D.2.1 milan9-gulfair-binary.names

```
!cst, cst
lexfirst: continuous.
lexsecond: continuous.
lexcommon: continuous.
commonnounfirst: continuous.
commonnounsecond: continuous.
```

```
commonnouncommon: continuous.
propernounfirst: continuous.
propernounsecond: continuous.
propernouncommon: continuous.
verbfirst: continuous.
verbsecond: continuous.
verbcommon: continuous.
adverbfirst: continuous.
adverbsecond: continuous.
adverbcommon: continuous.
adjectivefirst: continuous.
adjectivesecond: continuous.
adjectivecommon: continuous.
cardinalnumfirst: continuous.
cardinalnumsecond: continuous.
cardinalnumcommon: continuous.
lchnoun: continuous.
lchverb: continuous.
jcnnoun: continuous.
jcnverb: continuous.
resnoun: continuous.
resverb: continuous.
linnoun: continuous.
linverb: continuous.
wupnoun: continuous.
wupverb: continuous.
```

D.3 Trained data for ICSIBoost(shyp file)

D.3.1 A snippet from milan9-gulfair-binary.shyp

```
100

    1.000000000000  Text:THRESHOLD:lexcommon:

0.0000000000  0.0000000000

3.5099230536  -3.5099230536

1.1180199418  -1.1180199418

4.5000000000


    1.000000000000  Text:THRESHOLD:lexcommon:

0.0000000000  0.0000000000

1.4281655763  -1.4281655763
```

```

-0.3447050786 0.3447050786

2.5000000000


   1.000000000000  Text:THRESHOLD:lexsecond:

0.0000000000 0.0000000000

-0.7103295054 0.7103295054

0.5164424524 -0.5164424524

16.5000000000


   1.000000000000  Text:THRESHOLD:lchnoun:

0.0000000000 0.0000000000

0.2032092898 -0.2032092898

-1.1250090791 1.1250090791

2.7417063713
...
```

D.4 CST Bank Data Samples

D.4.1 CST Bank sample article in XML format (Cluster: Milan9 File: 1.docsent)

```
<?xml version='1.0'?>
<!DOCTYPE DOCSENT SYSTEM '/clair4/projects/mead307/stable/mead/dtd/docsent.dtd'>
<DOCSENT DID='1'>
  <BODY>
    <HEADLINE>
      <S SNO='1' PAR='1' RSNT='1'>CNN.com - Plane hits skyscraper in Milan - April 18, 2002</S>
    </HEADLINE>
    <TEXT>
      <S SNO='2' PAR='2' RSNT='1'>CNNenEspanol.com A small plane has hit a skyscraper in central Milan, setting the top floors of the 30-story building on fire, an Italian journalist told CNN.</S>
      <S SNO='3' PAR='3' RSNT='1'>The crash by the Piper
```

```
         tourist plane into the 26th floor occurred at 5:50 p.m
         . (1450 GMT) on Thursday, said journalist Desideria
         Cavina.</S>
      <S SNO='4' PAR='4' RSNT='1'>The building houses
         government offices and is next to the city's central
         train station.</S>
      <S SNO='5' PAR='5' RSNT='1'>Several storeys of the
         building were engulfed in fire, she said.</S>
      <S SNO='6' PAR='6' RSNT='1'>Italian TV says the crash put
          a hole in the 25th floor of the Pirelli building, and
          that smoke is pouring from the opening.</S>
      <S SNO='7' PAR='7' RSNT='1'>Police and ambulances are at
         the scene.</S>
      <S SNO='8' PAR='8' RSNT='1'>Many people were on the
         streets as they left work for the evening at the time
         of the crash.</S>
      <S SNO='9' PAR='9' RSNT='1'>Police were trying to keep
         people away, and many ambulances were on the scene.</S
         >
      <S SNO='10' PAR='10' RSNT='1'>There is no word yet on
         casualties.</S>
      <S SNO='11' PAR='11' RSNT='1'>U.N. envoy horror at Jenin
         camp U.S. bombing kills Canadians Chinese missiles
         concern U.S.   2002 Cable News Network LP, LLLP.</S>
      <S SNO='12' PAR='12' RSNT='1'>An AOL Time Warner Company
         .</S>
      <S SNO='13' PAR='13' RSNT='1'>All Rights Reserved. under
         which this service is provided to you.</S>
    </TEXT>
  </BODY>
</DOCSENT>
```

Bibliography

Mark Bauerlein. *The Dumbest Generation: How the Digital Age Stupefies Young Americans and Jeopardizes Our Future (or, Don't Trust Anyone Under 30)*. Jeremy P. Tarcher/Penguin, 2008. ISBN 1585426393. 1

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, 1998. URL `http://ilpubs.stanford.edu:8090/361/`. 2

Ziqiang Cao, Chengyao Chen, Wenjie Li, Sujian Li, Furu Wei, and Ming Zhou. Tgsum: Build tweet guided multi-document summarization dataset. *CoRR*, abs/1511.08417, 2015. URL `http://arxiv.org/abs/1511.08417`. 1, 3

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. Improving multi-document summarization via text classification. *CoRR*, abs/1611.09238, 2016. URL `http://arxiv.org/abs/1611.09238`. 3.1

Jean Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996. URL `https://www.aclweb.org/anthology/J96-2004`. 4.3.3

Yan-Min Chen, Xiao-Long Wang, and Bing-Quan Liu. Multi-document summarization based on lexical chains. pages 1937 – 1942 Vol. 3, 09 2005. ISBN 0-7803-9091-1. doi: 10.1109/ICMLC.2005.1527262. 2.2.1

Jackie Chi Kit Cheung and Gerald Penn. Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1233–1242, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P13-1121`. 2.2.2

Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association* *for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1012. URL `https://www.aclweb.org/anthology/N16-1012`. 2.1.3

Jason A. Cohen. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological bulletin*, 70 4:213–20, 1968. 4.3.3

John M. Conroy and Hoa Trang Dang. Mind the gap: Dangers of divorcing evaluations of summary content from linguistic quality. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 145–152, Manchester, UK, August 2008. Coling 2008 Organizing Committee. URL `https://www.aclweb.org/anthology/C08-1019`. 4.3.3

Yogita K. Desai and Prof. Prakash P. Rokade. Multi document summarization: Approaches and future scope. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 5(3), June 2015. 4

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`. 5

Jun'ichi Fukumoto. Multi-document summarization using document set type classification. In *NTCIR*, 2004. 2.2.1

Pierre-Etienne Genest and Guy Lapalme. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 354–358, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P12-2069`. 2.2.2

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 40–48, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1117575.1117580. URL `https://doi.org/10.3115/1117575.1117580`. 2.2.2

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`. 2.1.3

Anthony J. Hayter. *Probability and statistics for engineers and scientists.* Duxbury [u.a.], Pacific Grove, Calif., 2. ed edition, c2002. ISBN 0534386695 | 0-534-38669-5 | 9780534386696 | 978-0-534-38669-6. XII, 916 S, Ill, 25 cm. 4.3.3

Graeme Hirst and David St-onge. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An Electronic Lexical Database*, 305, 10 1995. 5

Mohamad Ali Honarpisheh, Gholamreza Ghassem-Sani, and Seyed Abolghasem Mirroshandel. A multi-document multi-lingual automatic summarization system. In *IJCNLP*, 2008. 2.2.2

Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of the International Conference on Research in Computational Linguistics*, 10, 10 1997. 4

Heng Ji, Benoit Favre, Wen-Pin Lin, Dan Gillick, Dilek Hakkani-Tur, and Ralph Grishman. Open-domain multi-document summarization via information extraction: Challenges and prospects. 07 2013. doi: 10.1007/978-3-642-28569-1_9. 2.2.2

Neha R. Kasture, Neha Yargal, Neha Nityanand Singh, Neha Kulkarni, Vijay Mathur, and Vijita Mathur. A survey on methods of abstractive text summarization. 2014. 2.1.2, 2.1.2

Atif Khan and Naomie Salim. A review on abstractive summarization methods. *Journal of Theoretical and Applied Information Technology*, 59:64–72, 01 2014. 2.1, 2.1.2, 2.1.2, 2.1.2

Bill Kovach and Tom Rosenstiel. *The Elements of Journalism: What Newspeople Should Know and the Public Should Expect.* Three Rivers Press, revised edition, 2014. ISBN 0804136785. 1

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977. 4.3.3

Claudia Leacock and Martin Chodorow. *Combining Local Context and WordNet Similarity for Word Sense Identification*, volume 49, pages 265–. 01 1998. 4

Yann LeCun, Y Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521: 436–44, 05 2015. doi: 10.1038/nature14539. 2.2

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W04-1013`. 4.1.4

Dekang Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304. Citeseer, 1998. 4

De-Xi Liu, Yan-Xiang He, Dong-Hong Ji, and Hua Yang. A novel chinese multi-document summarization using clustering based sentence extraction. volume 2006, pages 2592 – 2597, 09 2006. doi: 10.1109/ICMLC.2006.258855. 2.2.1

Fei Liu and Yang Liu. From extractive to abstractive meeting summaries: Can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 261–264, Suntec, Singapore, August 2009. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P09-2066`. 2.2.2

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv:1801.10198 [cs]*, 2018. URL `http://arxiv.org/abs/1801.10198`. 2.1, 4.1.1, 5

Elena Lloret and Manuel Sanz. Analyzing the use of word graphs for abstractive text summarization. *Proceedings of the First International Conference on Advances in Information Mining and Management (IMMM)*, 01 2011. 2.2.2

Elena Lloret, Laura Plaza, and Ahmet Aker. The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52:101–148, 2018. 4.3.2, 4.3.3

Rafael Margarido, Thiago A. S. Pardo, Gabriel Antonio, Vinícius B. Fuentes, Rachel Aires, Sandra Aluisio, and Renata Fortes. Automatic summarization for text simplification: Evaluating text understanding by poor readers. *Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, 10 2008. doi: 10.1145/1809980.1810057. 2.1

Kathleen McKeown and Dragomir R. Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 74–82, New York, NY, USA, 1995. ACM. ISBN 0-89791-714-6. doi: 10.1145/215206.215334. URL http://doi.acm.org/10.1145/215206.215334. 2.2.1

Kathleen R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Barry Schiffman, and Simone Teufel. Columbia multi-document summarization: Approach and evaluation. In *In Proceedings of the Document Understanding Conference (DUC01*, 2001. 2.2.1

Christian M. Meyer, Margot Mieskes, Christian Stab, and Iryna Gurevych. DKPro agreement: An open-source Java library for measuring inter-rater agreement. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 105–109, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14-2023. 4.3.3

Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-3252. 2

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://www.aclweb.org/anthology/K16-1028. 2.1.3, 4.2.1

Ani Nenkova and Kathleen McKeown. *A Survey of Text Summarization Techniques*, pages 43–76. Springer US, Boston, MA, 2012. ISBN 978-1-4614-3223-4. doi: 10.1007/978-1-4614-3223-4_3. URL https://doi.org/10.1007/978-1-4614-3223-4_3. 2.1.1

Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. Technical report, Microsoft Research, 2005. 3

Kimberly Neuendorf. *The Content Analysis Guidebook*. 01 2002. 4.3.3

B.M.M. Othman, M. Haggag, and M. Belal. A taxonomy for text summarization. *Information Science and Technology*, 3:43–50, March 2014. 1, 2.1, 2.1

Ramesh Pandita. Information pollution, a mounting threat: Internet a major causality. *Journal of Information Science Theory and Practice*, 2(4):49–60, Nov 2014. 1

Emily Pitler, Annie Louis, and Ani Nenkova. Automatic evaluation of linguistic quality in multi-document summarization. pages 544–554, 09 2010. 4.3.3

Dragomir Radev, Jahna Otterbacher, and Zhu Zhang. CSTBank: Cross-document Structure Theory Bank. http://tangra.si.umich.edu/clair/CSTBank, 2003. 4.1.3

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. MEAD - a platform for multidocument multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL `http://www.lrec-conf.org/proceedings/lrec2004/pdf/757.pdf`. 4.1.4

Dragomir R. Radev. A common theory of information fusion from multiple text sources step one: Cross-document structure. In *SIGDIAL Workshop*, 2000. 4.1.3

Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Comput. Linguist.*, 24(3):470–500, September 1998. ISSN 0891-2017. URL `http://dl.acm.org/citation.cfm?id=972749.972755`. 2.2.1

Dragomir R. Radev and Zhu Zhang. Cross-document relationship classification for text summarization. 2004. 4.1, 4.1.3, 4.1.3, 4.1.3, 4.1.3, 4.3, 4.1.3, 4.1.3

Juan Ramos. Using tf-idf to determine word relevance in document queries. 01 2003. 1

Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-363-8, 978-1-558-60363-9. URL `http://dl.acm.org/citation.cfm?id=1625855.1625914`. 4

Tony Rogers. Why don't young people read the news?, 2017. URL https://www.thoughtco.com/why-dont-young-people-read-the-news-2074000. 1

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015. 2.1.3, 2.1.3

Dipti Yashodhan Sakhare and Dr. Shyamala Rajkumar. Abstractive multi-document summarization : An overview. 2014. 1, 2.2

Judith D. Schlesinger, Dianne P. O'Leary, and John M. Conroy. Arabic/english multi-document summarization with classy - the past and the future. In *CICLing*, 2008. 2.2.1

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *ACL*, 2017. 2.1.3, 4.2.1, 4.3, 4.3.1, 4.3.4

Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, and Chandan K. Reddy. Neural abstractive text summarization with sequence-to-sequence models. *CoRR*, abs/1812.02303, 2018. 1, 2.1.3

Wired Stuff. Wikipedia creators move into news, 2004. URL https://www.wired.com/2004/11/wikipedia-creators-move-into-news/. 3.1

Ladda Suanmali and Naomie Salim. Literature reviews for multi-document summarization. 2008. 1, 2.2, 2.1

Allen Wastler. Newspaper bane: Nobody reads the stories, 2013. URL https://www.cnbc.com/id/100952247. 1

Aaron Weiss. The unassociated press, 2005. URL https://www.nytimes.com/2005/02/10/technology/circuits/the-unassociated-press.html. 3.1, 3.1

Wikimedia. Wikinews:mission statement, 2015. URL https://en.wikinews.org/wiki/Wikinews:Mission_statement. 3.1

Wikipedia contributors. Wikinews — Wikipedia, the free encyclopedia, 2005. URL https://en.wikipedia.org/w/index.php?title=Wikinews&oldid=27867383. [Online; accessed 8-August-2019]. 3.1, 3.1

Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. doi: 10.3115/981732.981751. URL `https://doi.org/10.3115/981732.981751`. 4, 5

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13:55–75, 08 2018. doi: 10.1109/MCI.2018.2840738. 2.1.3

Junlin Zhang, Le Sun, and Quan Zhou. A cue-based hub-authority approach for multi-document text summarization. pages 642 – 645, 01 2005. ISBN 0-7803-9361-9. doi: 10.1109/NLPKE.2005.1598815. 2.2.1