

Studies on Development of Resources and Methods for Extractive and Abstractive Summarization

Thesis Submitted by
Monalisa Dey

Doctor of Philosophy (Engineering)

Department of Computer Science & Engineering
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India

2025

JADAVPUR UNIVERSITY
KOLKATA-700032, INDIA

INDEX NO. 125/17/E

1. Title of the Thesis:

Studies on Development of Resources and Methods for Extractive and Abstractive Summarization

2. Name, Designation & Institution of the Supervisor/s:

Dr. Dipankar Das
Assistant Professor,
Department of Computer Science & Engineering,
Jadavpur University,
Kolkata - 700 032, India

3. List of Publication:

(A) Journal Publications:

- i. **M. Dey**, S. Mahata and D. Das. 2023. Exploring Summarization of Scientific Tables: Analysing Data Preparation and Extractive to Abstractive Summary Generation. *International Journal for Computers & Their Applications*, Vol. 30, No. 4, pp. 412–424.
- ii. **M. Dey**, S. Mahata, A. Mondal and D. Das. 2024. ECTI Transactions on Computer and Information Technology.

(B) Book Chapter Publications:

- i. **M. Dey** and D. Das. 2020. A deep dive into supervised extractive and abstractive summarization from text, In *Data Visualization and Knowledge Engineering: Spotting Data Points with Artificial Intelligence (2020)*, pp. 109–132.

(C) **Conference/Workshop/Symposium Publications :**

- i. **M. Dey**, S. Mandi and D. Das. 2018. Summarization of Table Citations from Text. In *Proceedings of the 15th International Conference on Natural Language Processing (ICON)*, pp. 35–42.
- ii. **M. Dey**, A. Mondal and D. Das. 2018. How to Analyze the Overall Performance of a Student: Strong or Weak. In *IEEE 5th International Congress on Information Science and Technology (CiSt)*, pp. 1–6.
- iii. **M. Dey**, A. Mondal and D. Das. 2017. JUNLP at IJCNLP-2017 Task 3: A Rank Prediction Model for Review Opinion Diversification. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 138–142.
- iv. A. Pramanik, S. Mandi, **M. Dey**, and D. Das. 2017. SciSumm 2017: Employing Word Vectors for Identifying, Classifying and Summarizing Scientific Documents. In *BIRNDL@ SIGIR*, pp. 94–98.
- v. **M. Dey**, A. Mondal, and D. Das. 2016. NTCIR-12 MOBILECLICK: Sense-based Ranking and Summarization of English Queries. In *The 12th NTCIR Conference Evaluation of Information Access Technologies*, pp. 138–142.

(D) **Miscellaneous:**

- i. A. Mondal, **M. Dey** and E. Cambria. 2023. An annotation system of a medical corpus using sentiment-based models for summarization applications. In *Computational Intelligence Applications for Text and Sentiment Data Analysis*, pp. 163–178.
- ii. **M. Dey**, A. Mondal, S. Mahata, D. Sarkar .2022. Breast Cancer Classification Using Deep Convolutional Neural Networks. In *Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing* pp. 179–187.
- iii. S. Mahata, A. Mondal, **M. Dey**, D. Sarkar. 2022. Sentiment Analysis using Machine Translation. In *Applications of Machine Intelligence in Engineering*, pp. 371–377.
- iv. A. Mondal, S. Mahata, **M. Dey**, and D. Das. 2021. In Proceedings of the Sixth Social Media Mining for Health (SMM4H) Workshop and Shared Task. In *In Proceedings of the Sixth Social Media Mining for Health (SMM4H) Workshop and Shared Task*, pp. 135–137.

4. **List of Patents:**

None

5. **List of Presentations in International Conferences:**

- **M. Dey**, S. Mandi and D. Das. 2018. Summarization of Table Citations from Text. In *Proceedings of the 15th International Conference on Natural Language Processing (ICON)*, pp. 35–42.
- **M. Dey**, A. Mondal and D. Das. 2018. How to Analyze the Overall Performance of a Student: Strong or Weak. In *IEEE 5th International Congress on Information Science and Technology (CiSt)*, pp. 1–6.

Statement of Originality

I, Monalisa Dey, registered on 10th March 2017, do hereby declare that this thesis entitled **"Studies on Development of Resources and Methods for Extractive and Abstractive Summarization"** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies. All information in this thesis has been obtained and presented in accordance with existing academic rules and ethical conduct. I declare that, as required by these rules and conduct, I have fully cited and referred all materials and results that are not original to this work. I also declare that I have checked this thesis as per the "Policy on Anti-Plagiarism, Jadavpur University, 2019", and the level of similarity as checked by iThenticate software is 4%.

Signature:

Monalisa Dey

Monalisa Dey

Certified by the supervisors (signature and seal)

1. Dipankar Das, 04/03/2024

ASSISTANT PROFESSOR
Dept. of Computer Sc. & Engg.
JADAVPUR UNIVERSITY
Kolkata-700032

Certificate from the Supervisors

This is to certify that the thesis entitled “**Studies on Development of Resources and Methods for Extractive and Abstractive Summarization**” submitted by **Ms. Monalisa Dey**, who got her name registered on March 10th, 2017, for the award of Ph.D (Engg.) degree of Jadavpur University is absolutely based upon her own work under the supervision of **Dr. Dipankar Das** and that neither her thesis nor any part of the thesis has been submitted for any degree or any other academic award anywhere before.

Signature of the Supervisors

Dipankar Das - 04/03/2024

Dr. Dipankar Das ASSISTANT PROFESSOR
Dept. of Computer Sc. & Engg.
JADAVPUR UNIVERSITY
Kolkata-700032
Department of Computer Sc. & Engg., Jadavpur University,
Kolkata – 700 032, India

Acknowledgements

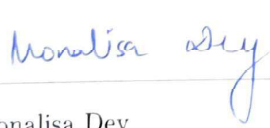
Research is often described as a methodical endeavor aimed at acquiring fresh insights. This journey towards new knowledge is greatly facilitated by receiving appropriate guidance and support. Hence, I wish to express my gratitude to those individuals who have aided me along the way.

First and foremost, I am grateful to the divine for granting me the strength to complete this thesis.

Next, I extend my heartfelt thanks to my family members for their unwavering support, especially during moments of self-doubt. From a young age, they instilled in me a belief that diligence and sincerity lead to success. This belief provided solace during times of uncertainty. I also owe a debt of gratitude to my mentor and supervisor, Dr. Dipankar Das, whose guidance was instrumental in completing my Ph.D. dissertation. The importance of mentorship in research cannot be overstated, and I am thankful for Dr. Das's unwavering support and the resources provided.

Lastly, I would like to acknowledge the invaluable assistance of my colleagues at my workplace: Dr. Sainik Kumar Mahata, Dr. Darothi Sarkar, Dr. Anupam Mondal, and Apurba Paul. Their constant support in managing the pressures of this endeavor made it feasible for me to pursue this dissertation.

Signature:

 04/03/24

Monalisa Dey

Abstract

The exponential growth of information accessible on the World Wide Web has posed a significant challenge in processing and summarizing vast textual content. This has led to the development of automatic text summarization systems to condense lengthy documents into manageable forms while retaining crucial information. Our research endeavors to tackle various challenges in automatic text summarization, focusing on both resource and model development.

Recognizing the unique nature of tables as a form of data representation, we embarked on exploring table summarization systems. However, table summarization faces a scarcity of large and diverse datasets for effective model training, unlike other natural language processing tasks such as machine translation or sentiment analysis. To overcome this hurdle, we delved into methods for crafting structured datasets tailored specifically for summarizing tables, thereby facilitating the development of robust summarization systems. Our work introduces several iterations of gold standard table summarization datasets. Additionally, we proposed the performance analysis dataset, further enriching our resources for generating student performance summaries. These datasets undergo meticulous annotation to ensure they provide relevant information for training effective summarization models.

In the domain of extractive summarization, we put forth novel methods, including rule-based and template-based approaches, to effectively summarize tabular data. These methodologies enhance the quality and relevance of extracted information, preserving essential content within the summaries. Following this, we developed a sense-based model aimed at effectively ranking and summarizing English queries to align with the objectives of the NTCIR shared task. Leveraging sentiment lexicons and tabulation-based methodologies, our system yielded positive outcomes, demonstrating its ability to reduce user interaction with mobile devices while extracting relevant information.

Furthermore, we conducted a study on summarizing scientific articles, as evidenced by our participation in the SCISUMM 2017 shared task. Drawing on our expertise in

extractive summarization, we devised methods to identify significant information within scientific literature. These endeavors contributed to enhancing knowledge dissemination and facilitating information retrieval within academic communities.

Subsequently, we explored extractive summarization using word embedding techniques, aiming to leverage word vector semantic representations to enhance the coherence and informativeness of extracted summaries. These models were trained and tested on open source gold standard datasets like CNN/Daily mail.

In the realm of abstractive summarization, we leverage deep learning methods such as LSTM and RNN architectures to craft coherent summaries. Through extensive training and testing on reliable open-source datasets, we ensure the accuracy and relevance of our abstractive summarization models. Moreover, we introduce a novel strategy using one-word abstractive summaries derived from our proposed performance analysis datasets, offering concise insights into student achievement.

A significant contribution of this thesis also lies in proposing and evaluating innovative table-based summarization systems that leverage extractive summaries to generate abstractive summaries. Noteworthy among these models are the "Fine-Tuned T5" model, a transformer-based approach fine-tuned on our custom dataset, a sequence-to-sequence encoder-decoder model, and the "SETA" model, an attention-based encoder-decoder model focused on similarity. These models undergo rigorous evaluation using standard metrics, showcasing their ability to produce coherent and informative summaries from table-based content.

Moreover, our research extended to the field of ranking in natural language processing, where we proposed novel models for query ranking tasks. By participating in collaborative tasks such as the NTCIR-12 shared task and the IJCNLP-17 Review Opinion Diversification task, we developed sense-based ranking systems and feature classifier based models for ranking queries.

Contents

Statement of Originality	i
Approval	i
Acknowledgements	v
Abstract	vii
Title of the Thesis	xi
List of Tables	xviii
List of Figures	xx
1 Introduction	1
1.1 Natural Language Processing	1
1.2 Information Retrieval	2
1.3 Summarization	3
1.3.1 Different Types of Summarization	4
1.3.2 Applications of Summarization	4
1.4 Challenges and Motivation	4
1.5 Thesis Overview	6
1.5.1 Dataset Preparation	7
1.5.2 Extractive Summarization	7
1.5.3 Abstractive Summarization	8
1.5.4 Extractive to Abstractive Summarization	9
1.5.5 Ranking	9
1.6 Contributions	10

2	Dataset Preparation	13
2.1	Introduction	13
2.2	Survey	15
2.3	Table Summary Dataset Preparation	17
2.3.1	Summary Extraction	19
2.3.2	Data Annotation and Guidelines	20
2.3.3	Evaluation : Agreement among Annotaters	22
2.4	Continuous Evaluation Dataset Preparation	22
2.4.1	Data Collection	23
2.4.2	Annotation and Agreement Analysis	23
2.5	State of the Art Dataset Description	25
2.5.1	NTCIR-12	25
2.5.2	SCISUMM-17	27
2.5.3	IJCNLP-2017	29
2.5.4	CNN	30
2.5.5	Opiniosis	31
3	Extractive Summarization	33
3.1	Introduction	33
3.2	Survey	34
3.3	Extractive Table Summarization System	36
3.3.1	Experimental Dataset	37
3.3.2	Model 1 – Rule Based System	37
3.4	Model 2 – Template based system	39
3.4.1	TF-IDF Based System	40
3.4.2	Transition Point Based System	40
3.4.3	Experiment and Results	42
3.5	Model 3 – Sense Based System	47
3.5.1	Experimental Dataset	49
3.5.2	IUnit Summarization	50
3.5.3	Evaluation	52
3.6	Model 4 – SCISUMM: Word Vector Based System	53
3.6.1	Experimental Dataset and Preprocessing	54
3.6.2	System Framework	54
3.6.3	Evaluation	57

3.7	Model 5 – Embedding Based System	58
3.7.1	Experimental Dataset	59
3.7.2	Word Embeddings with Weighted Mean Vectors	59
3.7.3	System Framework	61
3.7.4	Evaluation	63
3.7.5	Discussion	63
3.8	Conclusion	64
4	Abstractive Summarization	67
4.1	Introduction	67
4.2	Survey	69
4.3	Model 1 - Deep Neural Network Based System	71
4.3.1	RNN encoder decoder	71
4.3.2	Experiments and Discussion	72
4.4	Model 2 - Performance Analysis Based System	75
4.4.1	Experimental Dataset	76
4.4.2	Identification of Subject Classes/Categories	76
4.4.3	Strong-Weak Classification	78
4.4.4	Experiments and Results	78
4.5	Observation	79
5	Extractive to Abstractive Summarization	83
5.1	Introduction	83
5.2	Survey	86
5.3	Model 1 – Fine Tuned T5 Model	88
5.3.1	Experimental Dataset	88
5.3.2	System framework	89
5.4	Model 2 – Seq-to-Seq Model	91
5.4.1	Experiments and Results	93
5.5	Model 3 – SETA	93
5.5.1	Experimental Dataset	94
5.5.2	Dataset Quality Validation	94
5.5.3	Input Summary Selection	97
5.5.4	Vectorisation of the summaries	98
5.5.5	Sentence embedding subspace	98

CONTENTS

5.5.6	Encoder-Decoder Model with Attention	99
5.5.7	Experiment and Results	100
5.6	Observations	102
6	Ranking	105
6.1	Introduction	105
6.2	Model 1 – Sense Based Ranking	106
6.2.1	Experimental Dataset	106
6.2.2	IUnit Ranking	107
6.2.3	Evaluation	108
6.3	Model 2 – Opinion Review Ranking	109
6.3.1	Experimental Dataset	109
6.3.2	Subtask Description and Extractiong of Feature	110
6.3.3	Modules Building	111
6.3.4	Evaluation	111
6.4	Conclusion	113
7	Conclusion	115
	Bibliography	119

List of Tables

2.1	Statistics of Data	21
2.2	Experimental Dataset Statistics	24
2.3	Validating the Experimental Dataset	25
2.4	iUnits for English Query MC2-E-0008 :”Genghis’	27
2.5	Dataset Overview	30
2.6	Dataset Statistical Info	31
3.1	Dataset statistics	38
3.2	An inter annotator confusion matrix	43
3.3	Average Metrics Values	44
3.4	Automatic evaluation scores for summary templates	46
3.5	Summary template validation	46
3.6	Comparison between rule-based and TF-IDF-Unigram approach where both annotators have agreed	47
3.7	iUnit Text Corresponding to iUnit ID’s	51
3.8	Wup_similarity scores between intents and iUnits	52
3.9	Task 1 System Evaluation	57
3.10	Task 2 System Evaluation	58
3.11	Rogue - 2 scores	63
4.1	Parameters used in the seq to seq Model	73
4.2	Ablation study (F-measure) for overall performance class assignment	79
4.3	Comparison of subject class assignment	80
4.4	Evaluation metrics for overall student performance	80
5.1	Comparison between T5 and seq-to-seq model	93
5.2	Inter annotator agreement analysis	95
5.3	Extended Dataset Statistics	96

LIST OF TABLES

5.4	An inter annotator agreement analysis to validate the dataset	96
5.5	Automated Evaluation	100
5.6	Human Evaluation Results	101
6.1	Sense Difference Scores	107
6.2	Run File for Ranking	108
6.3	iUnit ranking system performance evaluation	109
6.4	Query Classification	110
6.5	A comparison of all submitted modules of subtask-B	111
6.6	Comparison of Subtask A participants	112
6.7	Subtask B assessment output	113
6.8	Assessment results of module for subtask-C.	113

List of Figures

2.1	Captions as Abstractive Summary	19
2.2	Reference as Abstractive Summary	20
3.1	Process of caption identification and relevant sentence extraction	37
3.2	Inter-annotator BLEU scores for rule-based approach	44
3.3	Inter-annotator ROUGE scores for rule-based approach	45
3.4	Inter-annotator BLEU scores for TF-IDF unigram Approach	47
3.5	Inter-annotator ROUGE scores for TF-IDF unigram Approach	48
3.6	Inter-annotator BLEU scores for TF-IDF bigram Approach	49
3.7	Inter-annotator ROUGE scores for TF-IDF bigram Approach	50
3.8	Overall System Framework	50
3.9	Mean M-measure for Four Query Types	53
4.1	An outline of the recommended classes and the subjects they would cover. .	77
5.1	Architecture of the seq2seq model used to generate abstractive summaries from extractive summaries	91
5.2	SETA – Architecture	97
6.1	Ranking Framework	107
6.2	Mean nDCG values for 4 Query Types at Varying Thresholds	108
6.3	Mean Q measure values for 4 Query Types at Varying Thresholds	109

Chapter 1

Introduction

1.1 Natural Language Processing

NLP or natural language processing, aims to give computers the same level of comprehension as humans in both spoken and written languages. It combines statistical, machine learning, and deep learning models with computational linguistics, which involves modeling human language using rules. These techniques have made it possible for computers to analyse text or audio data and understand the meaning that is conveyed through spoken or written language, including the intents and feelings of the speaker or writer.

NLP is everywhere. Software applications that can interpret text between languages, react to voice commands, and quickly summarise massive amounts of text—sometimes even in real time—are powered by natural language processing. Speech-to-text dictation apps, digital assistants, chatbots for customer support, voice-activated GPS units, and other consumer conveniences are likely examples of applications that use natural language processing.

It is highly challenging to build software that accurately ascertains the intended meaning of written or verbal material since human language is so confusing. All the peculiarities of human language, including homophones, sarcasm, idioms, metaphors, exceptions to usage rules and syntax, and changes to sentence structure, require years to learn. To make natural language-driven apps usable, programmers must first accurately recognise and comprehend how to construct those applications. The availability of extensive datasets and resources is a crucial component of NLP research. Massive language models have been trained using corpora like as the Books Corpus and Wikipedia, which have been essential in allowing these models to capture a wide range of linguistic trends and semantic correlations.

Natural language processing has advanced significantly in recent years due to the introduction of deep learning approaches, especially with neural networks (Young et al., 2018). Across a wide range of natural language processing problems, deep learning algorithms like transformers, convolutional neural networks (CNNs), including recurrent neural networks (RNNs) have demonstrated impressive performance, frequently outperforming conventional techniques. In language comprehension and generation tasks, for example, models like as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) have demonstrated state-of-the-art performance (Devlin and et al., 2023; Radford et al., 2019).

Several NLP jobs analyse human text and voice data to help the machine understand the data it is processing. Some of them are:

- **Automatic Summarization**, is the task of summarising by using a machine to create a clear, brief summary that captures the essential details and primary ideas of a longer document, like an article, report, or webpage (Nenkova and McKeown, 2011).
- **Speech Recognition**, is the accurate process of translating voice input into text. (Reynolds and Rose, 1995).
- **Part of Speech Tagging** is the process of determining a word's part of speech based on its usage and context (Van Halteren et al., 2000).
- **Word Sense Disambiguation** is the act of picking the meaning of a word from amongst its multiple alternative meanings. (Gliozzo et al., 2004).
- **Named Entity Recognition (NER)** (Carreras et al., 2003) recognises phrases or words that serve as helpful entities. The context of the sentence leads NER to identify "India" as a place and "Jack" as a name.
- **Co-Reference Resolution** is the process of detecting if the two phrases refer to the same entity (Clark and Manning, 2016).
- **Natural Language Generation** is the method of transforming structured information into natural language (Reiter and Dale, 1997).

1.2 Information Retrieval

A vital aspect of today's systems for managing information is information retrieval (IR), which includes methods and strategies for effectively extracting pertinent facts from massive

amounts of data. It is essential to many different fields, including e-commerce sites, digital library services, web engines for searching, and enterprise search tools (Manning et al., 2008). Information retrieval aims to provide users with relevant information by matching queries with appropriate files or materials within a collection.

In conventional information retrieval systems, user queries are processed and pertinent materials are retrieved through methods including indexing, querying, and sorting (Baeza-Yates and Ribeiro-Neto, 2011). To determine document relevance and order search results, these systems usually use vector space models and term frequency-inverse document frequency (TF-IDF) methods. However, modern IR systems confront new obstacles due to the exponential growth of digital data and the growing complexity of user queries.

Arguably the most essential elements of information retrieval systems is summarization. Information retrieval systems can efficiently convey the most pertinent information to consumers, saving time and effort in the information consuming process, by producing succinct summaries of large documents or datasets. In digital libraries, content aggregation platforms, and web search engines alike, summarization is essential to improving the usability and accessibility of these systems.

Information retrieval has undergone a radical transformation thanks to recent developments in machine learning, deep learning, and natural language processing (Jurafsky and Martin, 2019). Major improvements regarding accuracy and relevance have been made possible by methods like transformer-based designs, and deep learning structures (Devlin et al., 2019).

1.3 Summarization

The amount of information that is available has significantly increased since the World Wide Web was established. Numerous repositories, including book, news, and scientific article archives, contain a tremendous volume of textual content. It is almost hard for a human to process and, moreover, summarise the vast amount of information available. As a result, users must spend a lot of time searching for the information they need. Moreover, a huge lot of unnecessary or redundant content is included in the final documents, making comprehension even more challenging. As a result, it is imperative to summarise and condense the textual resources. Furthermore, it is nearly impossible to summarise by hand. The solution is Automatic text summarization (Allahyari et al., 2017). The goal of automatic text summarization is to retain most of the content while condensing a lengthy document into a readable form (Tas and Kiyani, 2007).

1.3.1 Different Types of Summarization

There are two primary methods for accomplishing automatic text summarization: extractive and abstractive. Traditional extractive summarising techniques involve selecting and rearranging words or sentences that already exist in the source text. However, the rich context and meaning of the original content are not always preserved by these procedures. On the other hand, abstractive summary techniques, as outlined by [Allahyari et al. \(2017\)](#), seek to circumvent these limitations by producing summaries that both condense and preserve the coherence and important components of the material. Abstractive summarization is considered more challenging than extractive summarization because it requires real-world data and semantic class analysis ([Sunitha et al., 2016](#); [Fabbri et al., 2019](#)). Sentences that preserve the main ideas and concepts from the original text with the least amount of repetition are acceptable outcomes for both types of summaries. Sentences should also be coherent and consistent in order to preserve the text’s sense, even when they are long ([Sunitha et al., 2016](#)).

1.3.2 Applications of Summarization

Summarization systems are vital to many applications of NLP, or natural language processing, because they provide ways to extract and condense important information from massive amounts of text input. These systems are extensively employed in several fields to provide effective information retrieval, understanding, and decision-making procedures. summarising systems are used in NLP for tasks like content creation, news aggregation, social media monitoring, text summarising, and text generation tasks as well (([Bogdanova and Loukachevitch, 2020](#); [Nenkova and McKeown, 2011](#)). They help users make decisions and acquire knowledge more rapidly by making it possible for them to easily understand the main ideas of lengthy papers or publications. Additionally, summarising methods help to improve information accessibility, especially in situations where consumers are overloaded with textual material. Through the automatic generation of clear and logical summaries, these systems enable users to efficiently sift through large volumes of information.

1.4 Challenges and Motivation

One of our main focus was on generating table based summarization systems as s summarising tables presents a special difficulty in the field of natural language processing,

namely the lack of high-quality training dataset. Most summarization tasks lack huge and diverse corpora, in contrast to other NLP tasks such as machine translation or sentiment analysis, which have accessibility to large, curated datasets. This lack of data is a significant difficulty for training robust summarization systems, as models need large amounts of labelled text to learn well. Furthermore, the absence of standardised evaluation measures and benchmarks complicates comparing summarization performance across models and datasets. As a result, academics frequently resort to manual annotation or synthetic data generation approaches to address these restrictions and progress the field of summarization in NLP. This leads us to our first research question,

How can we create high-quality annotated and structured datasets for summarising tables?.

Moreover, text summarization in natural language processing (NLP) presents various obstacles due to the complexities of human language and the wide range of textual data it handles. One key challenge is reducing significant details while keeping the essential ideas and value of the text in its entirety. Furthermore, summarising algorithms must manage language characteristics such as its context, style, and intent in order to provide clear and short summaries. Furthermore, these systems must adapt to a variety of document kinds, each with its own structural and linguistic features. This leads us to the research question,

Is it possible to design extractive summarization algorithms that can produce coherent and comprehensive summaries?

Although abstractive summary methods have an advantage over extractive methods despite their complexity, writing correct abstractive summaries is difficult, and academics have investigated numerous approaches and methodologies to overcome the obstacles of abstractive summarization. This led us to our third reasearch question,

In what ways might the application of deep learning algorithms produce abstractive summaries that are both informative and coherent?

Another key problem in this field involves creating abstractive summaries from extractive content. Researchers have increasingly concentrated on using extractive summarising outputs to improve the quality and fluency of abstractive summaries, despite the dis-

tinct benefits and drawbacks of each method. The justification for employing extractive summarising outputs to generate abstractive summaries stems from their complimentary nature. Extractive approaches, which choose and incorporate sentences directly from the source document, excel at keeping the original context and assuring factual accuracy. However, their summaries frequently lack coherence and consistency, as they may struggle to adequately restate or generalise knowledge. Abstractive techniques, on the other hand, allow you to interpret and paraphrase material to create summaries that closely resemble human language.

Furthermore, it should be highlighted that in the field of natural language processing, table-based summarisation is one area of summary development that has not gotten as much attention. The scarcity of acceptable datasets makes summarising tables a distinct difficulty. Unlike traditional text-based summarization, which can use enormous corpora, table-based summarization relies on structured data that is difficult to acquire. Tables often contain a wide range of information, including numerical data and categorical factors, necessitating the development of datasets capable of capturing this complexity. However, creating such datasets necessitates meticulous annotation and curation, which adds complexity to the process. This led us to our final research question,

How can tables in scientific articles be efficiently summarised by extractive-to-abstractive summarising systems?

We were also interested in learning about how these systems can enhance the effectiveness of summarization algorithms and make them more human-like. All of these challenges motivated us to look into these most difficult aspects of automatic text summarization.

1.5 Thesis Overview

In the current thesis, we made an effort to address the earlier posed research questions. To address the challenge of dataset scarcity, we were motivated to design gold standard datasets for table summarization as well as perform comprehensive validation tests in Chapter 2. We were motivated to examine and develop several extractive summarization methods in Chapter 3. Chapter 4 highlights our contribution in developing novel abstractive summarization systems. In Chapter 5 we outline our contributions to the field of extractive to abstractive summarization, and in Chapter 6, we present our contribution in the domain of ranking as an important part of information retrieval.

1.5.1 Dataset Preparation

Chapter 2 presents our efforts on extensive dataset preparations. As discussed in the previous sections, summarising tables presents a special difficulty in the field of natural language processing because there aren't many appropriate datasets designed for this particular task. Table-based summarization necessitates structured data that is not easily accessible, in contrast to standard text-based summary, which can rely on large corpora such as news items or literary works. To fill this significant gap, we have set a goal of creating extensive, annotated datasets specifically tailored for table-based summarization tasks.

Moreover, in this chapter we have also reported how we have actively engaged with a variety of standard open-source datasets in our quest to improve the capabilities of extractive and abstractive summarization algorithms. These datasets cover a wide range of concepts and text genres, which enables us to train and assess our models in a variety of contexts.

Furthermore, we have actively participated in shared tasks like IJCNLP 2017 (Singh et al., 2017b), NTCIR-12 (Kato et al., 2016), and SCISUMM 2017 (Jaidka et al., 2016) in order to develop extractive summarization systems. This has allowed us to access additional datasets and support collaborative research efforts as discussed in this chapter. Through the utilisation of a wide range of diverse datasets such as DUC, Opinosis, and CNN/Daily Mail, we aim to improve the effectiveness, resilience, and usability of our summarization models.

1.5.2 Extractive Summarization

Chapter 3 presents our novel contributions aimed at advancing the field of extractive summarization, after we thoroughly examined several extractive summarization techniques. We started our investigation by carefully concentrating on tabular data summaries from scholarly publications. A rule-based technique and a template-based approach were the two distinct methods we proposed. Through careful evaluation, which included validation processes and dataset quality assessment, we were able to increase the quality of our dataset. The relevance and compactness of the dataset were enhanced by this refining technique, which made it possible to extract information from tables more effectively.

Next, we developed a sense-based model to effectively rank and summarise English queries in order to meet the NTCIR shared task objective. By employing sentiment lexicons and tabulation-based methodologies, our system produced positive outcomes,

showcasing its capacity to reduce user engagement with mobile devices while extracting pertinent information.

In addition, we conducted a study on scientific article summaries, as demonstrated by our involvement in the SCISUMM 2017 shared task. Using our knowledge of extractive summarization, we created methods to find important information in scientific literature. In academic communities, these initiatives aided in streamlining knowledge transfer and information retrieval.

Next, we looked at extractive summarization using word embedding techniques. Our goal was to use word vector semantic representations to increase the extracted summaries' coherence and informativeness. Our tests and analysis provided thorough insights into the effectiveness of different strategies and shed light on how different techniques impact the quality of summaries.

Moreover, we also employed rigorous methodologies for system evaluation, such as automated evaluation metrics like BLEU and ROUGE and inter-annotator agreement-based validation. These evaluations gave us important information on the effectiveness and performance of our methods, enabling us to make well-informed decisions for further development and improvement.

1.5.3 Abstractive Summarization

Chapter 4 explores the area of abstractive summarization. The chapter's first section focuses on using deep learning methods, such as LSTM and RNN approaches, to create logical and educational summaries. Research is conducted to investigate the use of the encoder-decoder RNN architecture, a versatile sequence-to-sequence framework, for the implementation of abstractive summarization. To guarantee the accuracy and relevance of the outcomes, these models undergo extensive training and testing on dependable open-source datasets such as DUC and CNN/Daily Mails.

The chapter also discusses challenges encountered during the design of this deep learning-based model, such as vocabulary size constraints and issues with loss computation and output production.

The chapter also highlights how important it is to develop a method for analysing student achievement. As a result, a brand-new strategy was created, utilising one-word abstractive summaries generated from a carefully chosen performance analysis dataset. This innovative technique has the ability to give educators and other stakeholders the critical knowledge they need to make well-informed decisions and implement effective

intervention strategies by distilling complex numerical data on student performance into concise and insightful insights.

Furthermore, comprehensive experiments are conducted on several datasets to evaluate the effectiveness and resilience of the proposed solutions. Metrics that are especially created to fulfil the objectives of every methodology for model training, dataset selection, and assessment are included in the experimental framework.

1.5.4 Extractive to Abstractive Summarization

In Chapter 5, we mainly focus on developing table based summarization systems, that use extractive summaries as a foundation to create abstractive summaries of tables in scientific papers. To do this, initially we propose an further extended version of our gold standard dataset. Next we propose three models for summarising table data. First model is the fine tuned T5 model which is based on transformers. We fine tune it using our own customised extended dataset. The second model we propose is a sequence-to-sequence model. Finally, the third model proposed by us is the SETA model, an attentional based encoder-decoder model based on similarity. Each of these models generates an abstractive summary after choosing an appropriate extractive summary for each table, based on the extractive to abstractive summary generation philosophy. The findings imply that our systems are able to generate coherent summaries.

1.5.5 Ranking

In the discipline of Natural Language Processing (NLP), ranking problems are crucial because they facilitate the effective organisation and retrieval of data from massive textual databases. Ranking is a crucial component of many information retrieval systems, with uses ranging from question-answering and document summarising to online search and recommender systems.

Chapter 6 details our participation in two significant collaborative tasks, primarily related to query ranking. We explore the NTCIR-12 shared task, one of the major tasks in the field of extractive summarization, in the first section of the chapter. Particularly, the two subtasks of the NTCIR-12 task are the ranking and summary of web-based query outputs. In this chapter, the ranking subtask was our main focus. We discussed about the model we proposed along with the process of testing the model using metrics provided by the organisers.

Our participation in the IJCNLP-17 Review Opinion Diversification challenge is covered

in the next section. Ranking a product’s top k reviews from an extensive amount of reviews is the task’s objective. This makes it easier to compile a brief overview of each opinion shared in those reviews. The task is divided into three subtasks A, B, and C. Selecting the top-k evaluations based on a variety of criteria, including representativeness, helpfulness, and exhaustiveness, is the aim of each task respectively. This chapter reports our endeavor to solving subtasks A, B, and C, by creating three modules with an emphasis on representativeness, exhaustiveness, and utility.

1.6 Contributions

Tables frequently contain a wide range of information, from numerical data to categorical factors, so it is critical to create datasets that represent this complexity. But the process of creating such databases is made more difficult by the need for careful annotation and validation. This inspired us to take on the task of creating the best annotated and structured datasets possible for table-based summaries. More precisely, we have suggested two table-based summary datasets in **Chapter 2**. The purpose of the first dataset is to summarise tables seen in research publications. A student’s performance is evaluated continuously using the second dataset, which identifies their strengths and weaknesses in different categories of subjects. Moreover, in our endeavor for developing extractive and abstractive summarization systems, we have also extensively engaged with various standard open-source datasets thereby allowing us to test our models on diverse data.

We began our investigation into tabular data extraction from scientific papers in **Chapter 3** of our study. Firstly we proposed an extended version of our gold standard table summarization dataset that we introduced in Chapter 2. Next, We developed two models, a rule-based model and a template-based model, to address the challenge of table summarization.

Moving on to our NTCIR shared task contribution, we created a sense-based model that is designed to efficiently summarise English queries. Our approach showed promise in reducing user participation with mobile devices while extracting relevant data by utilising tabulation-based algorithms and sentiment lexicons. In order to

We then extended our research to include summarising scientific papers, as evidenced by our involvement in the shared SCISUMM 2017 assignment. Here, we used our experience with extractive summarization to create models that are specifically made to extract important information from scientific papers.

Additionally, we integrated word embedding techniques into extractive summarising

and proceeded to construct an extractive summarization model. Our goal was to use semantic representations contained in word vectors to improve the coherence and informativeness of extracted summaries.

We outline our major contributions to the field of abstractive summarization in **Chapter 4**. Our first work is focused on leveraging deep learning methods, specifically LSTM and RNN approaches, to construct abstractive summarization models. These models are validated and tested on reliable open-source datasets, such as DUC and CNN/Daily Mails, ensuring the relevance and accuracy of our findings. Furthermore, we introduce a novel strategy that leverages one-word abstractive summaries, extracted from our own meticulously constructed performance analysis dataset, to summarize a student's strengths and weaknesses.

We primarily concentrate on creating table-based summarising systems in Chapter 5, which build abstractive summaries of tables in scientific articles using extractive summaries as a base. First we propose an extended table summarization dataset so as to include more training data. Next we propose three models for extractive to abstractive summarization task. The first model proposed is the transformer-based, fine-tuned T5 model. We use our own customised expanded dataset to fine-tune it. A sequence-to-sequence model is the second one we propose. Finally the third model developed by us is the SETA model. This is an attentional based encoder-decoder model based on similarity. Based on the extractive to abstractive summary generation philosophy, each of these models selects an acceptable extractive summary for each table before producing an abstractive summary.

Finally in **Chapter 6** we propose two novel contributions to field of ranking in natural language processing. Specifically, this chapter describes our participation in two important collaborative tasks that are mostly concerned with query ranking. First we propose a ranking model by participating in the NTCIR-12 shared task. This is sense based ranking system. The second model developed by us is an outcome of our participation in the IJCNLP-17 Review Opinion Diversification task which mainly concentrates on ranking the top-k reviews for a particular product. We propose three ranking models based on usefulness, representativeness and exhaustiveness of the reviews.

Chapter 2

Dataset Preparation

2.1 Introduction

Summarising tables presents a distinct problem in natural language processing because of the limited availability of appropriate datasets. Unlike traditional text-based summarization, which can leverage extensive corpora such as news articles or literary works, table-based summarization requires structured data that is not as readily available. Tables often contain diverse information ranging from numerical data to categorical variables, making it essential to develop datasets that capture this complexity. However, creating such datasets requires meticulous annotation and curation, further complicating the process.

Recent admirable works like TableBERT ([Liu et al., 2022](#)) and TabularBERT ([Zhang et al., 2021a](#)) highlight the significance of using pre-trained models, especially tailored to table understanding and generation. The methods show how using extensive datasets can enhance the efficiency of table-based summarization systems. Even so, the lack of publicly available datasets is a major obstacle to progress in this field.

The dearth of datasets for table-based summarization stems from several factors ([Dey et al., 2023](#)). Firstly, existing corpora primarily focus on textual data and lack the necessary structure for training models specifically tailored to tables. Moreover, tables vary in formats and functions across different fields, requiring the development of datasets tailored to certain domains to enhance the reliability and applicability of summarization algorithms. Additionally, the manual annotation required to create high-quality datasets is labor-intensive and time-consuming, deterring researchers from embarking on such endeavors ([An et al., 2021](#)).

Addressing the need for datasets in table-based summarization requires a concerted effort from the research community. Collaborative initiatives to collect and annotate table

data from diverse sources could significantly accelerate progress in this field. Furthermore, leveraging techniques such as data augmentation and transfer learning may help mitigate the challenges associated with limited datasets, enabling the development of more effective summarization models. By prioritizing the creation and sharing of high-quality datasets, researchers can unlock the full potential of table-based summarization and pave the way for advancements in natural language understanding and generation.

This motivated us to take on the challenge of developing gold standard annotated and structured datasets for table based summary. More specifically, in this work we have proposed two table based summary datasets. The first dataset is for summarizing tables in scientific publications. The second dataset is a continuous evaluation dataset to measure a students performance based on their strong and weak genres.

Moreover, in our endeavor for developing extractive and abstractive summarization systems, we have also extensively engaged with various standard open-source datasets. These datasets include a broad spectrum of text kinds and topics, which enables us to train and assess our models in a variety of scenarios. For instance, we have utilized the CNN/Daily Mail dataset¹, which comprises news articles paired with human-generated summaries, providing valuable resources for extractive summarization tasks. Additionally, we have explored the DUC (Document Understanding Conference) dataset², renowned for its collection of news articles and corresponding abstracts, serving as a benchmark for evaluating abstractive summarization techniques. Furthermore, our work extends to Opinosis dataset (Ganesan et al., 2010), which offers opinionated text data along with succinct summaries, facilitating research in sentiment-based summarization approaches.

Additionally, for developing extractive and abstractive summarization systems, we have actively participated in shared tasks such as IJCNLP 2017 (Singh et al., 2017b), NTCIR-12 (Kato et al., 2016), and SCISUMM 2017 (Jaidka et al., 2016), thereby accessing additional datasets and contributing to collaborative research efforts. These shared tasks provide valuable opportunities to work with domain-specific datasets and evaluate our models in real-world scenarios. For example, our involvement in the IJCNLP 2017 shared task enabled us to access datasets tailored to specific natural language processing challenges, fostering innovation and benchmarking our summarization approaches against those of other researchers. Similarly, participation in NTCIR-12 and SCISUMM 2017 shared tasks exposed us to datasets curated for tasks related to sense based ranking , information retrieval and scientific document summarization, broadening the scope of our research

¹<https://www.kaggle.com/datasets/gowrishankarp/newspaper-text-summarization-cnn-dailymail>

²<https://www-nlpir.nist.gov/projects/duc/guidelines/2002.html>

and facilitating cross-domain learning. By leveraging these open-source datasets, including CNN/Daily Mail, DUC, and Opinions, our goal is to improve the efficiency and resilience of our summarization systems while advancing the field of natural language processing research.

The rest of the chapter is arranged as follows. The most recent developments in this field of study are covered in Section 2.2. Section 2.3, demonstrates the entire process of a gold standard data preparation along with its annotation and validation. Next Section 2.4 describes the development of a performance analysis dataset preparation and finally Section 2.5 introduces the details of the standard open source datasets that we have used for developing summarization systems.

2.2 Survey

Table content summarization systems have various potential applications, such as summarising patient information based on symptoms, predicting weather from daily reports, summarising Wikipedia infoboxes, and analysing sports based on score tables and many more. However, previous studies indicate that a major challenge lies in identifying an appropriate corpus for training, testing, and evaluating table summarization systems. This section delves further into the current literature on table-based summarization datasets

A large-scale dataset named FINDSum was recently proposed by [Liu et al. \(2023\)](#) for long text and multi-table summarization. It is based on 21,125 yearly filings from 3,794 corporations and includes two subsets for tabulating each company’s operational and liquidity outcomes. There are also some additional recent datasets for table-to-text creation or table summarization available. [Liang et al. \(2009\)](#) introduced WEATHER-GOV, a dataset for generating weather forecasts based on common tables. The world state is condensed in this dataset via records that combine measurements during particular time periods. There are 29,528 scenarios in all, with an average of 36 records and 28.7 words each scenario.

In their paper, [Lebret et al. \(2016\)](#) introduced WikiBio, which represents a table by concatenating field, position, and word embeddings. Their dataset is also far more diversified, with 400,000 words compared to a few hundred for Weathergov or Robocup. However, this dataset, too, only includes brief (single-sentence) generations with few records each generation. Early progress on these datasets may not be enough to fully evaluate text production capabilities at a document scale.

[Chen and Mooney \(2008\)](#) presented another dataset in their paper, ROCOBUP, which

represents robotic football events. For the purpose of developing semantic parsers from ambiguous supervision, the dataset was compiled, comprising both unlabeled and gold-labeled data. The noisy dataset was created by temporally matching a stream of football events from a robotic football match with human comments about the game. However, the dataset was not adequate in size.

Wiseman et al. (2017) introduced a novel dataset intended for extensive table summarization. This dataset consists of summaries of basketball games of NBA from two separate sources, as well as box and line-score tables. The first source, ROTOWIRE, offers rather lengthy game summaries specifically designed for fantasy basketball players. Conversely, the second dataset, SBNATION, is made up of summaries written by fans for other fans. SBNATION’s language is informal and often veers away from the statistical material, making it both significantly harder and significantly larger.

Furthermore, extracting tables from text documents is a difficult operation. The job in (Ng et al., 1999) was completed by identifying the row, column, and table border. These are categorised as three distinct classification problems, and they rely on training texts that are examples of correctly identified table boundaries, columns, and rows by human annotators. Classifiers are then constructed from the training samples using machine learning methods, one classifier for each sub-problem.

A machine learning-based method has been utilised in (Wang and Hu, 2002) to classify tables in HTML documents as authentic or non-authentic. A collection of innovative features has been established that mirror the structure and substance of tables. Given the very non-homogeneous nature of the features in this task, the decision tree classifier is employed for table detection. Additionally, they conducted experiments using support vector machines, which exhibit the greatest results in text classification. But plenty of test data were misclassified by this system.

Tengli et al. (2004) describes an automated table extraction method that discovers lexical variations from training examples, responds to non-exact pairings amongst labels using a vector space method, and takes advantage of formatting cues in the semi-structured HTML tables.

Tables in (Chen et al., 2000) are extracted from extensive HTML texts. There are five components in this task: table sorting, recognition, comprehension, hypertext processing, and finally result display. Heuristic rules are used in the table filtering module in order to filter out implausible instances. The content of the cells is how the table recognition module recognises the table. The attribute-value connection in a table can be interpreted

either column-wise or row-wise by the the table interpretation module. Finally, the table containing a succession of attribute-value pairs is displayed by the presentation of results module.

Much work has been done in the topic of table summarization, as was previously discussed. All of these approaches, however, do not address the problem of creating a corpus for scientific table summarising systems. The task of creating a gold standard corpus with extractive and abstractive summaries of scientific tables has been tackled in our study.

2.3 Table Summary Dataset Preparation

In order to communicate information, authors include a variety of images in their paper, including tables, graphs, and flowcharts. These components are necessary when presenting results, workflows, or system flows. Over time, the importance of obtaining information from these components has grown. Particularly tables frequently include important experimental findings and concepts. Thus, scholars can save a great deal of time and effort by being able to study more data by being able to understand table information without having to read the entire manuscript. In fact, summarization tools can assist readers automatically extract important information in situations like this. Even though there are so many advantages, studies have demonstrated that it is difficult to locate appropriate data for these systems’ training, testing, and evaluation.

Our motivation for addressing this issue led us to begin our research with the extraction of a set of table-content summaries using NLP-driven algorithms in two different formats: extractive and abstractive summaries. Preparing the data, extracting information, creating modules, and validating the outcomes are all steps in the process. To put it simply, we take lines from the article that represent the content of each table to determine its extractive summary, and we take the original captions associated with each table to get the abstractive summary.

For creating this gold standard data corpus, we have taken into consideration and implemented workable solutions for the challenges that follow:

Dataset Development: To address this challenge, we collected 499 distinct tables from 200 computer science journals that cover a variety of areas, including machine translation, named entity recognition, and sentiment analysis, among others. Since the majority of articles are only available in PDF format, we converted them to text for additional

analysis using PDFTextStream³.

Generating an abstract summary from the table caption: The fact that captions are written in a multitude of formats across multiple domains makes them challenging to extract. In order to get around this problem, we found that a table’s caption sentence has FOUR components; *<TEXT>*, *<TABLE>*, *<DELIMITER>*, and *<INTEGER>*. We suggest using any statement that satisfies the previously specified pattern to distinguish a caption from the other phrases. The caption may then be used as an abstractive synopsis of the table.

Generating an extractive summary from table reference text Although captions provide perspectives into the table’s data, they may not always provide enough detail for readers to completely comprehend the topic. To circumvent this limitation, we extracted those parts of text from the papers which references the table . With a few slight variations in the pattern, we employed a similar approach as in the prior challenge. Sentences around the reference sentence often give accurate contextual information about how the table is used, as we have seen. Hence, we also extracted and recorded these contextually relevant sentences.

Validating the quality of the dataset The system’s assessment procedure is split into two sections: dataset quality evaluation and the accuracy scores of the summaries. We have used two annotators—our system and A1, a manual annotator—to validate the first section. The inter-annotator agreement scores are then examined using the Cohen’s Kappa agreement analysis method. We used a sentiment-based similarity technique to judge the quality of the output summaries, and the result was a similarity score between the summaries produced by the system and the reference summaries, which are also recognised by A1. The prepared corpus can be regarded as a gold standard dataset as well. This is the case since we used the author’s own scientific publication’s captions and text to create the corpus.

Representing the data as a structured corpora In order to tackle this difficulty, we have created an annotated corpus that includes details about an assortment of tables and the attributes that are associated with them, such as the number of rows, columns, abstractive summary, and so on.

The contributions of this task include addressing the previously mentioned challenges and delivering a structured corpus with annotations and output summaries. This corpus can be used by researchers as a benchmark dataset for summarising table content.

³<http://snowtide.com/PDFTextStream>

2.3.1 Summary Extraction

We prepared the corpus using scientific publications retrieved from digital libraries. In order to collect the papers, we downloaded over 200 papers from the ACL ⁴ database, encompassing 20 various topics in the field of computer science. Each downloaded paper typically contains an average of about 202 sentences, not including the title, names or headings. Table 2.1 displays the statistics of our corpora. The stages below provide an overview of the dataset construction process.

Classifier	Accuracy	P	R
DTree	66.744	66.846	67.382
LogR	67.907	67.089	69.806
SVM	69.069	66.277	80.317

Table 4: Single document input classification Precision (P), Recall (R) and F Score (F) for difficult on DUC'01 and '02 (Total 432 examples) divided into 2 classes based on the average coverage score (217 difficult and 215 easy inputs)



<Paper ID =1> <Table ID =4> <Abstractive summary> =277 Table 4: Single document input classification Precision (P), Recall (R) and F Score (F) for difficult on DUC'01 and '02 (Total 432 examples) divided into 2 classes based on the average coverage score (217 difficult and 215 easy inputs) </Abstractive Summary> <Extractive Summary> = Multi-document task from the results in Table 4 it is evident that all three classifiers achieve accuracies higher then those for multi-document summarization </Extractive Summary> </Paper ID =1>

Figure 2.1: Captions as Abstractive Summary

Captions as Table Abstractive Summary: An effectively crafted caption clearly conveys the table information shown in a table. Depending on the writing style and domain, captions can be written in a variety of styles. It has been noted in multiple articles that a caption sentence comprises four elements. The terms are composed of <TABLE>, which stands for “Table,” and <INTEGER>, which stands for an integer that is associated with the document’s table number. A delimiter, such as a period or colon, specifies the punctuation mark at the end of the phrase and comes after the number. The last section is <TEXT>, which describes the contents of the table. A caption sentence is defined as a statement that follows this pattern and makes up the abstractive summary content of the table. This process is describe in Figure 2.1.

Relevant Sentences as Table Extractive summary: Studies have demonstrated that captions alone are insufficient to fully communicate an element to a reader, even though they effectively convey the content of a table. We have noticed that every table is cited in

⁴<https://aclanthology.org/>

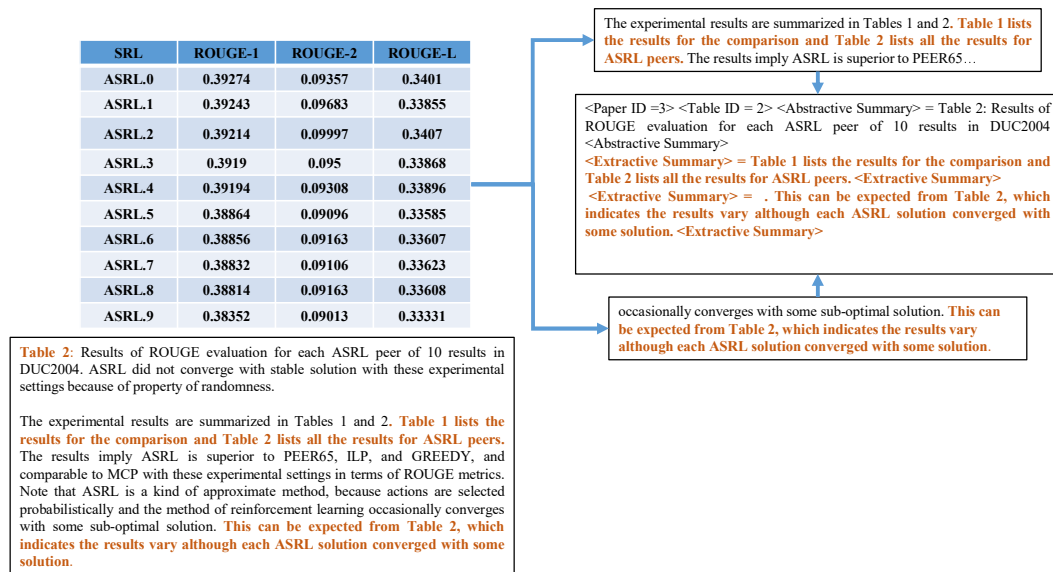


Figure 2.2: Reference as Extractive Summary

the paper at least once in order to address this problem. As a result, we have taken the table’s reference text out of the related scientific publication in order to have a deeper comprehension of it. Sentences were extracted from the document text as our initial step. We have used the identical methodology as outlined for caption extraction to find relevant sentences; the only distinction is that these sentences lack the delimiter. Furthermore, the phrases that follow a reference sentence in a document that references a table can be highly helpful in elaborating on the context of the table’s mention. In light of this, we have rated each sentence according to how close it is to the reference sentence and how far away it is from it. If the distance is within a predefined threshold length (+/-1) then the sentence is considered relevant and incorporated to the summary. This process is describe in Figure 2.2

2.3.2 Data Annotation and Guidelines

Annotation: After producing high-quality abstractive and extractive summaries based on the gold standard, an annotated dataset is compiled to establish an organised, clearly defined, and user-friendly corpus. To build an excellent data set for automatic review and evaluate the extractive and abstractive efficacy of our system-generated summaries, two different approaches were taken. Initially, an external computer science expert annotator was employed. The abstractive and extractive summaries in the dataset had to be manually found by the annotator. Furthermore, our system produced an output feature

Type	# Tables	Text Tables	Numeric Tables
Text Summarization	50	21	29
Machine Translation	55	19	36
Machine Learning	45	22	23
Names entity Recognition	51	25	26
QnA	60	25	35
Text Classification	44	21	23
Speech Recognition	31	19	12
Sentiment Analysis	42	19	23
Text Segmentation	64	21	43
Word Sense Disambiguation	61	29	32
#Total Papers: 200			

Table 2.1: Statistics of Data

file that contained features other than the summaries. Paper ID, table ID, row count, column attributes, row attributes, table type (text, numeric, or both), and row attributes are among the features.

Guideline: After our processing of 200 scientific publications in the field of computer science, the corpus is arranged into distinct folders for each paper. There are six different files within each of the folders. The files are as follows:

- CSV: This includes individual CSV files that describe the contents of each table in the paper.
- Summary: The files provide hand recognised extractive and abstractive summaries for each table in the study
- PDF_Document: This is the document in PDF format.
- Text_Document: This contains the paper in text format.
- XML_Document: This is the paper in XML format.
- Annotation: These are automatically generated descriptions of the feature files for all tables in the publication.

The corpus also has a *README* file that thoroughly explains every feature of all the files. A brief discussion of the evaluation method is provided in the next subsection.

2.3.3 Evaluation : Agreement among Annotators

We use a manual annotator named A1 to assist in our assessment. A1 and our system are given 100 unique documents each, selected randomly, and are required to independently differentiate between extractive and abstractive summaries. Each sentence in the summary is assigned a score of "1" if included, and "0" if excluded. We will assess the amount of agreement by examining 100 documents annotated by A1 and 100 documents produced by our proposed system. The papers are exchanged, and the annotators are asked to confirm their agreement (1) or disagreement (0) with the other system's output.

As a result, we had 200 papers total that both annotators had graded. Twenty tables with 4040 summary sentences for both extractive and abstractive summaries were chosen from this set. Next, we use **Cohen's Kappa coefficient** κ to calculate the agreement score between annotator A1 and the output generated by the system. This value is provided by

$$\kappa = \frac{Pr_a - Pr_e}{1 - Pr_e}, \quad (2.1)$$

The variable Pr_a represents the proportion of complete agreement between the two chosen annotators. Moreover, Pr_e represents the percentage expected by random chance, indicating a type of arbitrary consensus among annotators.

This agreement score yields $\kappa = 0.832$ and $\kappa = 0.813$ for abstractive and extractive summary, respectively. Higher κ indicates a significant agreement. This experiment was designed to determine how well the suggested method identified the document's table content summary.

2.4 Continuous Evaluation Dataset Preparation

Through a variety of applications, NLP approaches have been employed to improve educational technology during the previous few decades (Burstein et al., 2014). Personalised learning materials, automated scoring systems, and dialogue technologies for tutoring are a few examples of these applications (Nogaito et al., 2016; Schwind, 1988). In the education domain, research is also being conducted to establish a performance analysis system for students (Hearst, 2015; Schwind, 1988).

A few-hour topic tests are insufficient to assess a student's calibre; a performance analysis system is necessary. All year long, including extracurricular activities like physical education, artistic education, and social and personal traits, student assessments must be

conducted. The absence of domain specialists, such as instructors, in the design process of such automated systems is posing a significant challenge to researchers today. There are other issues as well, like the absence of parental support and the inaccessibility of data regarding student absenteeism in rural schools and colleges. We were therefore inspired to create a performance analysis system that uses NLP-based methodologies to help parents and instructors assess a student’s overall strengths and weaknesses.

2.4.1 Data Collection

Developing a performance monitoring system requires a well-defined, structured dataset. We were unable to uncover organised data from any past study in this area for use in our system design. To tackle the issue of insufficient data, five academic institutions provided us with a collection of relevant data depicted in the statistics as categories 1-5. This was then processed and our experimental data was ready to use.

In order to protect the privacy of the students, anonymous 2564 student result samples are included in the gathered dataset. After that, in order to accurately depict our experimental dataset, we eliminated any missing values and eliminated other types of noise, such as Unicode and white spaces. Ultimately, 2045 student result samples containing values from 50 individual students are included in the experimental dataset.

A team of educators has annotated the experimental dataset using two labels: the first pertains to the five distinct subject classes, and the second label concerns the student’s overall performance, specifically their strengths and weaknesses. It has been discovered that some courses/subjects may be found in many classes. To further develop and evaluate the proposed system, we partitioned the data set used for experimentation into two parts: the training and test data. The training dataset comprises 1500 samples (73.35%), while the remaining 545 samples are contained in the test dataset (26.65%). The dataset’s detail statistics are displayed in Table 2.2. Here capability denotes strength and limitation denotes weakness. The distribution of classes as well as the strengths and weaknesses for each student sample are also included in the table.

2.4.2 Annotation and Agreement Analysis

Policy for Data Annotation: This section outlines the manual annotating strategy employed to annotate the class for performance analysis with the guidance of multiple academics. This policy essentially includes 50 different disciplines for a student, along with their 5 subject classes along with 2 of the general ones. After that, we annotated the

CHAPTER 2. DATASET PREPARATION

	Category 1		Category 2		Category 3		Category 4		Category 5	
Total Data Items	529		527		493		479		536	
Disposable Data	116		122		96		81		104	
Experimental Data	413		405		397		398		432	
Training Data	303		297		291		292		317	
Test Data	110		108		106		106		115	
Subject Categories	Strength	Weakness	Strength	Weakness	Strength	Weakness	Strength	Weakness	Strength	Weakness
Soft Skills	104	309	93	312	67	330	90	308	145	287
Logic and Quantitative Potential	147	266	148	257	119	278	121	277	134	298
Basic Skills	203	210	223	182	194	203	199	216	216	215
Programming Skills	121	292	135	270	112	285	121	277	119	313
Specialized Knowledge	75	338	81	324	54	343	81	317	84	348
Overall Performance										
Strength	290		311		210		223		293	
Weakness	123		94		187		175		139	

Table 2.2: Experimental Dataset Statistics

experimental dataset using the procedures outlined below.

Step-1: A raw text file was created containing the entire data that was collected from three separate academic institutions. Every line of the text document contains data on every student of those institutions.

Step-2: Next we removed all noisy and erroneous data and created a pre-processed file for experimentation.

Step-3: Next, a specialised set of educators used their domain expertise for assigning labels namely topic classes and classes of overall performance to each of the pupil records.

Agreement Analysis: Additionally, with the help of two chosen professors, we employed an agreement analysis approach for validating the dataset for experimentation.

To do this, we calculate the agreement between annotators A_1 and A_2 using Cohen’s Kappa⁵ agreement analysis approach which is defined as follows [Viera et al. \(2005\)](#):

$$\kappa = \frac{Pr_a - Pr_e}{1 - Pr_e} \quad (2.2)$$

where the observed percentage of complete agreement between two annotators is denoted by Pr_a . Furthermore, Pr_e represents the percentage predicted by chance, suggesting a form of haphazard agreement among the annotators.

For every label listed, a separate agreement analysis was performed. Table 2.3 displays the agreement-based ratings provided by two annotators for the two labels. The κ scores derived from Equation 6.3 are 0.79 and 0.82 for the annotation labels, signifying almost perfect annotation for the dataset.

⁵https://en.wikipedia.org/wiki/Cohen's_kappa

#Samples: 2045			1st Annotator		κ score
			Yes	No	
2nd An-notator	<i>Subject Classes</i>	Yes	1624	61	0.79
		No	64	296	
	<i>Overall Performance Class</i>	Yes	1754	33	0.82
		No	45	213	

Table 2.3: Validating the Experimental Dataset

2.5 State of the Art Dataset Description

As discussed earlier, in our works, we extensively use open-source datasets like CNN/Daily Mail⁶, DUC, and Opinions. These datasets cover diverse domains and text types, aiding in model training and evaluation. CNN/Daily Mail dataset offers news articles with human-generated summaries for extractive summarization tasks. DUC provides news articles and abstracts, serving as an abstractive summarization benchmark. Opinions contains opinionated text with succinct summaries, facilitating sentiment-based summarization research. Additionally, we participate in shared tasks like IJCNLP 2017, NTCIR-12, and SCISUMM 2017. These tasks provide access to more datasets and collaborative research opportunities. Participation in IJCNLP 2017 grants access to specific NLP challenge datasets, fostering innovation. NTCIR-12 and SCISUMM 2017 expose us to curated datasets for information retrieval and scientific document summarization. Overall, our engagement with diverse datasets and shared tasks enhances our summarization system development. The following sections provide in-depth descriptions of various open source datasets.

2.5.1 NTCIR-12

The objective of the NTCIR-12 (Kato et al., 2016) task is to summarise and evaluate English queries. The principal obstacle of this undertaking was to design and implement a system that could reduce the amount of manual labour required by users to interact with mobile devices while retrieving pertinent information in response to specified queries. The organisers supplied the data in the form of information units (iUnits). Every individual iUnit delineates a relevant inquiry that is accompanied by additional data such as type or category, relevance, sense, and knowledge-based relationships. Two subtasks comprise the assignment: ranking and summarization. The objective of the ranking subtask is to determine which iUnits are most significant in relation to a given query. The output for

⁶<https://github.com/deepmind/rc-data>

the summarization subtask must be formulated as a two-layered model: the initial layer must discern the significant iUnits, while the subsequent layer shall compile said iUnits into a summarised output that corresponds to the query.

Dataset: The shared task provided us with a test collection developed for the NTCIR-12 MobileClick-2 Task, containing Training, Test, and evaluation data for both English and Japanese subtasks. The NTCIR-12 MobileClick test collection consists of a hundred queries written in English and a hundred in Japanese. Using inquiries retrieved from a Wider Planet toolbar journal by actual users, the organisers translated the inquiries into Japanese and English. The prevalent queries were categorised into the following: QA, CELEBRITY, DEFINITION, and LOCAL, with each category comprising a predetermined quantity of queries.

The CELEBRITY category includes names of celebrities, while LOCAL involves landmarks, facilities, or entities with geographical constraints. DEFINITION consists of ambiguous terms that are frequently defamed, whereas QA consists of queries utilising natural language. The participants were furnished with a collection of iUnits derived from the top 500 documents returned by the Bing search engine for each query. On average, 418 documents were retrieved for English queries, while 442 documents were retrieved for Japanese queries.

For MobileClick task, iUnits, which are pertinent, atomic, and dependent units of data, were utilised as the unit of information. They are considered relevant if they provide useful factual information to the user, atomic if they cannot be further divided without loss of meaning, and dependent if they rely on other iUnits to be relevant. The total number of iUnits per query is 23.8 for English adding upto 2317 and 41.7 for Japanese queries, adding upto 4,169. Table 2.4 providing a snapshot of iUnits for English queries is presented below.

In addition, the notion of intent was introduced to signify particular interpretations of ambiguous inquiries or aspects of faceted queries, respectively. Intentions were considered and used as potential links to the second layer in the iUnit summary subtask in order to assess the relevance of iUnits. The mean number of intents per query obtained was 4.48 for English subtasks and 4.37 for Japanese subtasks. The significance of each iUnit was evaluated by two assessors on a five-point scale, with global importance being derived from the probability and per-intent importance.

iUnit ID for Query MC2-E-0007	iUnit Text
MC2-E-0008-001	born on the island of Corsica
MC2-E-0008-002	defeated at the Battle of Waterloo
MC2-E-0008-003	established legal equality and religious toleration
MC2-E-0008-004	an innovator
MC2-E-0008-005	absent during Peninsular War
MC2-E-0008-006	cut off European trade with Britain
MC2-E-0008-007	general of the Army of Italy
MC2-E-0008-008	one of the most controversial political figures
MC2-E-0008-009	won at the Battle of Wagram
MC2-E-0008-010	baptised as a Catholic

Table 2.4: iUnits for English Query MC2-E-0008 :”Genghis’

2.5.2 SCISUMM-17

The primary focus of the third CL-SciSumm shared task (Jaidka et al., 2016) was summarising scientific papers. This assignment was motivated by the fact that citations are an important tool for characterising scientific papers that have been utilised in information retrieval and summarization applications. Extending citation information could further benefit these applications. The objective of this work was to accomplish this by locating and categorising textual citation information, then using that information for summary. The following is the task description:

Given: A topic with a maximum of 10 Citing Papers (CPs) that all cite the Reference Paper (RP). The text spans, or citances, in each CP that are related to a specific citation to the RP have been identified.

Task 1A: Determine which text passages (cited text spans) in the RP most closely correspond to each citation. These can be as specific as a single sentence, a complete sentence, or up to five consecutive sentences.

Task 1B: From a predetermined list of aspects, determine which facet of the document each cited text span belongs to.

Task 2: Using the quoted text spans from the RP, create a summary of the RP. The summary shouldn’t be more than 250 words long. This additional job was optional.

Below is a discussion of the shared task dataset overview.

Dataset: The CL-SciSumm task used the same training data from the Pilot Task TAC ⁷

⁷<https://tac.nist.gov/2014/index.html>

2014, which was a component of the BiomedSumm Track at the Text Analysis Conference 2014. Ten sets from an entirely novel development corpus were used for system training in the shared task, while ten sets from a different test corpus were used for evaluation. For every set in every corpus, the job provided three different types of summaries:

Abstract: The writers of the research article wrote this,

Community summary: This is assembled from its citations’ reference spans.

And, a **summary generated by human annotators** participating in the CL-SciSumm annotation project.

The dataset was developed using the same process as the pilot task, which is stated below:

Development of Training Data: The training set for the track consists of twenty topics. Every topic has multiple components. Ten citation papers and a reference publication make up these components. Annotations of the citations from the articles that are cited are also supplied. Additionally, a synopsis of the reference work is provided. The reference paper’s discussion in the citing papers is included in this summary. It also takes into account the sections of the reference paper that the citing publications mentioned. Lastly, the reference paper’s abstract is also provided. Annotations and summaries from four distinct human annotators are included for each topic.

Interestingly, there were two differences in the citing paper (CP) selection between the training and test corpora. First off, when building the latter, the minimum quantity of CP required—three in the former—was raised to eight. Second, the former only allowed for a maximum of 10 CPs; however, the latter’s construction eliminated this cap, allowing for the provision of up to 60 CPs for a single RP. This was done in an effort to increase the number of citations, some of which might go into further information about the RP. The community overview would also result in a more comprehensive viewpoint as a result.

Annotation: Five postgraduate students from the University of Hyderabad, India, who study applied linguistics, annotated the development and test corpus. The participants were chosen from a group consisting of more than twenty-five individuals who had received training on how to use the Knowtator annotation package of the to annotate an RP and its CPs. The annotation group was given the task of locating citations to each RP in each of the CPs that were associated with it. The citation text, marker, text, and discourse facet were identified in the citing paper for every citation of the reference paper.

2.5.3 IJCINLP-2017

The objective of the IJCINLP-17 Review Opinion Diversification (RevOpiD-2017) ([Singh et al., 2017b](#)) is to provide a ranking of the topmost 'k' reviews of some selected products from a given set. This task aids in the identification of a concise output that encapsulates the total opinion expressed in the set of review. Three distinct subtasks comprise the task at hand; they are designated Subtask A, B and C.

Subtask A (Ranking of Usefulness): The objective is to rank the topmost 'k' reviews in a corpus of reviews pertaining to a specific product based on their predicted utility rating. In addition, redundancy among the ranked reviews must be penalised.

Subtask B (Ranking of Representativeness): The objective is to rank the topmost 'k' reviews from a corpus containing reviews pertaining to a specific products in order to optimise the representativeness of the ranked list. The ranking system ought to succinctly encapsulate the viewpoints articulated in the evaluations that were provided as input, striking a balance between novelty and diversity.

Subtask C (Exhaustive Coverage Ranking): The objective is to rank the topmost 'k' reviews pertaining to a specific product in a way that ensures the inclusion of the majority of widely recognised viewpoints while penalising redundant reviews within the ranked list. Similar to Subtask B with the exceptions listed as follows: Subtask B involves assessing the degree to which the ranked list accurately represents the most frequently held opinions in the review corpus in relation to the final ranking. Subtask C assesses the final ranking based on the extent to which the opinions in the final ranking provide comprehensive coverage.

Dataset: According to [McAuley and Leskovec \(2013\)](#), the opinion annotated dataset utilised in this shared task was obtained by extracting a subset of Amazon SNAP online reviews dataset. More than 34 million evaluations, encompassing more than 2 million products, were included in the initial SNAP dataset. A total of 85 products were selected from the given options, representing 12 distinct categories, and annotated with personal opinions. After that, the extracted data corpus was divided into three separate sections:

The training data, which was identical to the SNAP dataset, with the exception that it was a subset of the SNAP dataset. The statistics pertaining to the training data are presented in Table 2.5. The development data which comprised text review files and annotated opinion matrices pertaining to thirty products. An evaluation algorithm utilised these matrices to assess the performance of the systems that participated in Subtasks B and C (representing representativeness and excitement, respectively). And the test data,

which comprised exclusively the review text files of fifty products, which were devoid of any efficacy scores as well. We refrained from utilising the opinion matrices in order to assess final scores using the test data at hand.

Data Overview			
Category	Products	Reviews	Average Reviews per Product
Automotive	571	172,106	312
Baby	1,002	352,231	342
Beauty	1,010	316,536	326
Digital Music	469	145,075	310
Grocery	810	293,629	368
Health	1,100	357,669	358
Office	1,001	327,556	329
Patio and Lawn	861	263,489	316

Table 2.5: Dataset Overview

2.5.4 CNN

Dataset: The CNN/DailyMail ⁸ dataset comprises slightly more than 300,000 distinct news articles authored by personnel from CNN and the Daily Mail. The dataset is in the English language. The present iteration incorporates support for both extractive and abstractive summarization. However, its inception was focused on machine reading and comprehension, as well as abstract question responding. The dataset is structured as follows:

- **Data instances:** These consist of three strings: an identifier, a string representing the article, and a string representing the highlights. One may examine additional instances by utilising the CNN/Daily Mail dataset viewer.
- **Input Data Fields:** The fields are ID: A string comprising the SHA1 hash of the URL from which the story was retrieved, formatted in hexadecimal. Article: A string comprising the news article’s substance, and Highlights: A string comprising the author-written highlights of the article.
- **Initial Data Acquisition:** Highlighted sentences and news articles comprise the dataset. Articles function as context in the question-and-answer format, while entities

⁸https://huggingface.co/datasets/cnn_dailymail

are systematically concealed within highlighted sentences to generate Cloze-style inquiries.

Three subsets comprise the CNN/DailyMail dataset: train, validation, and test. Table 2.6 presents statistical information pertaining to the dataset in Version 3.0.0.

Split	Instances per Split
Train	287,114
Test	11,480
Validation	14,368

Table 2.6: Dataset Statistical Info

2.5.5 Opinosis

The dataset was created in 2010 and consists of review sentences taken from user reviews on a given topic. Sample topics include “Toyota Camry performance” and “iPod Nano sound quality,” among others. There are 51 topics in total, with an average of 100 sentences per topic. The evaluations for different products were obtained from external sources, including Edmunds.com for automobiles, Tripadvisor for hotels, and Amazon.com. The dataset was applied to a project involving the artificial summarization of texts.

The dataset file includes gold standard summaries utilised in the summarising paper mentioned above. The authors have included scripts to assist with the summarization/evaluation tasks using ROUGE, in addition to the dataset ⁹.

⁹<https://github.com/kavgan/opinosis-summarization/blob/master/opinosis-dataset-documentation.pdf>

Chapter 3

Extractive Summarization

3.1 Introduction

With the introduction of the World Wide Web, there is a notable rise in the amount of data that is available. A great amount of textual content can be found in a variety of repositories, such as archives of books, news items, and scientific papers, and more. This sheer volume of information makes it nearly impossible for a human being to process and, even more, summarise it all. Users consequently have to invest a great deal of time in finding the information they require. Furthermore, resulting texts include a great deal of redundant or insignificant content making it even more difficult to comprehend. Thus, it becomes vital and crucial to summarise and condense the text resources. Moreover, the process of manually summarising is practically impossible. As a solution, the most essential component in resolving this issue is Automatic Text Summarization ([Allahyari et al., 2017](#)). Condensing a long document into a readable text while keeping the majority of the material is the aim of automatic text summarization ([Tas and Kiyani, 2007](#)). Currently, there are two types of text summary techniques: extractive and abstractive ([Gambhir and Gupta, 2017](#)).

Extractive summarization ([Liu et al., 2018](#)) ([Liu and Lapata, 2019](#)), is a critical task in natural language information retrieval, aiming to condense large amounts of text into concise summaries while preserving important information. The basic idea is to develop a summary, that appropriately summarises a text document d , which consists of M sentences, by selecting a portion of the document's sentences. In order to further this field of study, this chapter mainly examines and develops several extractive summarization methods and highlights our original contributions.

Our research begins by focusing on summarizing tabular data extracted from scientific papers. Tables in scientific literature contain structured information such as experimental results and statistical analyses. We introduce two approaches for summarizing this data:

a rule-based method and a template-based approach. These methods aim to efficiently extract key information from tables to facilitate insights extraction from scientific literature.

Next, we demonstrate the NTCIR shared task, which emphasizes sense-based ranking and summarization. We propose a novel approach that combines sentiments with ranking strategies to generate informative summaries.

Next, in the SCISUMM, 2017 shared task, we focus on summarizing scientific articles. Leveraging our expertise in extractive summarization, we develop tailored systems to distill essential information from scientific papers effectively. These contributions aim to streamline information retrieval and promote knowledge dissemination within academic communities.

Additionally, we have explored the integration of word embedding techniques into extractive summarization to leverage semantic representations encoded within word vectors. By incorporating these techniques, we aim to enhance the coherence and informativeness of extracted summaries.

The rest of the chapter is structured as follows: Section 3.2 showcases the state of the literature, Sections 3.3, 3.4, 3.5, 3.6 and 3.7 describes a range of extractive summarization methods developed by us to address diverse challenges across various datasets and tasks. Finally 3.8, concludes the chapter briefly. In summary, by showcasing our contributions in this field, we aim to advance the field of extractive summarization and contribute to efficient information extraction and knowledge representation.

3.2 Survey

The process of extractive summarization entails choosing important phrases, paragraphs, and other components from the source content. The most important parts of the original text are highlighted in a succinct summary that is produced by combining these elements. Diverse algorithms have been put forth for extractive summarization, utilising distinct methods for ranking and extracting highly ranked sentences. Among these are statistical techniques based on word or sentence frequency. There are also graph-based methods which use ideas like centrality and community detection to treat the document as a graph of phrases. Similarly sentence meanings are the focus of latent semantic-based techniques (Rieger, 1991). This section examines all the current approaches to extractive summarization in detail.

The most popular model seems to be the graph-based models. They are extensively

used for nlp tasks like summarizing documents as they can efficiently describe document structure. One method combines external knowledge from Wikipedia via a bipartite graph structure (Sankarasubramaniam et al., 2014), which employs an iterative ranking algorithm similar to the HITS algorithm (Kleinberg, 1999). Another popular model which is also a graph-based method, LexRank (Erkan and Radev, 2004), determines sentence salience based on Eigen vector centrality, representing sentences as a graph with edges weighted by cosine similarity values. This utilization of graph structures enhances the selection of important sentences while ensuring coherence in the final summary. Other graph-based techniques based on TextRank include TopicRank (Bougouin et al., 2013) and PositionRank (Florescu and Caragea, 2017). Using a topic-modeling approach, Sentences with related subjects are grouped together by TopicRank, which then selects the most significant sentences from each cluster. In order to score sentences, PositionRank considers both the phrase frequencies in a biased PageRank and the distribution of term positions within a text.

Researchers have also investigated the fuzzy technique to tackling an extractive summarization challenge. Fuzzy logic techniques typically have four components: a defuzzifier, a fuzzifier, a fuzzy knowledge base, and an inference engine. These approaches take into account textual features like sentence length and similarity and input them into a fuzzy system (Kyoomarsi et al., 2008; Suanmali et al., 2009a). (Suanmali et al., 2009b) also suggested a fuzzy logic approach for this task, which uses pre-processing, feature extraction, and sentence ordering to ensure coherence. This inclusion of fuzzy logic allows for a more nuanced comprehension of textual features, which improves the summarising process.

Furthermore, concept-based techniques extract concepts from text by leveraging other knowledge bases such as HowNet (Wang et al., 2005). (Sankarasubramaniam et al., 2014) suggested a Wikipedia-based summary approach that employs graph structure, successfully incorporating external knowledge into the summarization process.

On the other hand, Latent Semantic Analysis techniques (Ozsoy et al., 2011; Mashechin et al., 2011) uncover underlying semantic structures of words and sentences, which are commonly used in text summarization without requiring outside or training data. A number of authors presented the LSA approach in (Gong and Liu, 2001). It entails representing a text as a matrix of terms and sentences that shows how frequently each word occurs in each phrase. The most important semantic elements of the document are then determined via singular value decomposition (SVD), ranked, and extracted. This tactic does have some drawbacks, though, especially when it comes to sentence choice. To

overcome them, (Steinberger et al., 2004) developed a semantic-based strategy employing LSA and more complex algorithms. Despite these benefits, however, the inclusion of SVD in LSA-based summarising approaches might make them computationally expensive, especially for larger texts (Yeh et al., 2005).

Finally, neural networks are used for extractive summarization, which offers a sophisticated way to detecting key sentences. One method employs a two-layer neural network with backpropagation trained on the RankNet algorithm (Svore et al., 2007). Another uses a three-layered feed forward neural network to learn the features of summary and non-summary phrases (Kaikhah, 2004). Authors in (Hingu et al., 2015) proposed an extractive method for summarising Wikipedia articles and ranking sentences utilising a model with neural networks. The use of neural networks improves the accuracy and efficiency of extractive summarization approaches.

The next few sections explore our novel contributions to the field of extractive summarization.

3.3 Extractive Table Summarization System

As discussed before, our journey towards generating extractive summarization systems begins by focusing on summarizing tables in scientific papers. Multiple research papers and articles inundating our lives pose difficulties in swiftly and effectively extracting pertinent information from non-textual elements such as tables, graphs, figures, and flowcharts. Tables, an integral part of scientific papers, provide an efficient medium to present intricate information concisely and in an organized manner. Nevertheless, tables present a challenge to conventional retrieval methods due to the fusion of content and presentation they embody. Therefore, the provision of having a system which provides a succinct summary of table data assumes a paramount role.

Initially the development of such systems faced a major challenge: the lack of appropriate corpora for training and evaluating summarization algorithms. This motivated us to develop a baseline gold standard dataset for table summarization the details of which is described in Section 2.3 and in subsection 2.3.1. A notable point in this regard is that for each table, the assumption was that for one abstractive summary there maybe multiple extractive summaries.

The upcoming sections 3.3.2 and 3.4, focus on initially developing a more extensive version of the previously created dataset and then proposing rule-based and template-based extractive summarization systems. The rule-based system provide us with the

most relevant and pertinent abstractive-extractive summary pair for each table in the dataset. Whereas, the template-based system develop summary templates representing the extractive summary of a table. These templates can be utilized by any NLP researcher to study and develop more accurate and coherent summaries and also to generate abstractive summaries.

Section 3.4.3 then demonstrates the experiments and results of our system and compares the performance of these two systems where we employ automatic as well as manual evaluation techniques to assess the quality of the summaries created by both the systems.

3.3.1 Experimental Dataset

As defined in Table 2.3, the baseline dataset comprised 499 distinct tables collected from over 200 computer science articles spanning different areas such as Sentiment analysis, named entity recognition, and Machine Learning, among others.

For generating the rule based system we were motivated to develop a more extensive version of the same. To construct this extensive corpus, we obtained 1,500 papers across distinct natural language processing areas. Each article has an average of approximately 250 sentences, not including titles, author names, and section headings. The basic steps are the same as described in sections 2.3.1 and shown in Figure 3.1. Table 3.1 shows the statistics of the collected dataset. Table types can be text or numeric. ESummary denotes extractive summary and ASummary denotes abstractive summary. $\#Eavg$ is the average number of extractive summaries present per table per paper in a particular paper type.

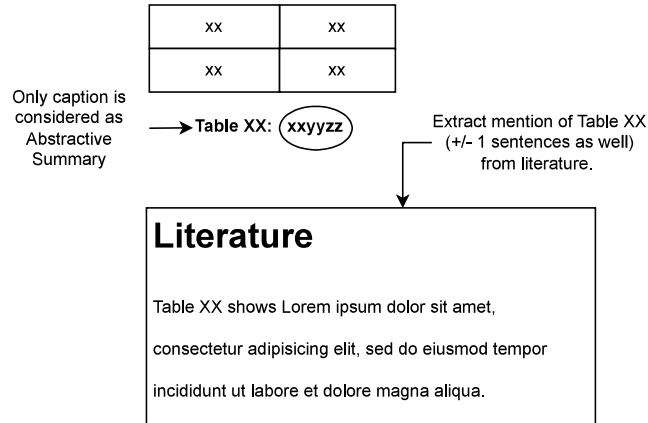


Figure 3.1: Process of caption identification and relevant sentence extraction

3.3.2 Model 1 – Rule Based System

After developing the dataset, we proposed a rule based system to select the most relevant extractive-abstractive summary pair. The details is described in the following sections.

Paper Type	#Tables	Type: Text	Type: Numeric	ESummary		ASummary
				#Eavg	Elen_avg	Alen_avg
Automatic Summary	510	130	370	3	16	11
Machine Learning	700	373	327	4	18	12
Machine Translation	420	150	280	3	16	10
Named Entity Recognition	789	553	236	2	16	14
Question Answering	553	120	433	3	15	13
Sentiment Analysis	421	125	296	2	14	14
Speech Recognition	700	432	286	5	13	13
Text Classification	567	265	302	3	15	15
Text Segmentation	700	432	268	2	13	13
Word Sense Disambiguation	650	324	326	1	11	13
Total no. of papers	1,500					

Table 3.1: Dataset statistics

Relevant Extractive Summary Selection: Extractive Summary Selection (ESS) is the process of selecting the most relevant extractive summary. It is crucial as it ensures that the model is trained on the best possible data and is more likely to produce accurate and informative summaries. Since our main aim is to ensure that the best quality extractive summary is selected for further works, we have used standard quality assessment tools like ROUGE (ESS_R), BLEU (ESS_B) and LEXRANK (ESS_L) and used a majority voting technique between them for selecting the most relevant extractive summaries.

It must be remembered that in the upcoming sections, the abstractive and the extractive summaries are considered as the reference and generated summaries respectively.

ESS_B : The BLEU score evaluates the accuracy of translations or summaries produced by computers in comparison to one or more references produced by humans. For every table i , the BLEU score between the abstractive summary and the extractive summaries for table i is calculated.

ESS_R : The effectiveness of automated summarization is evaluated using the ROUGE method. It determines how comparable the produced and the reference summary are dependant on the n-grams overlap and their respective frequencies. The difference between the abstractive summary and the relevant extractive summaries for each table i is measured by the ROUGE score.

ESS_L : LexRank [Erkan and Radev \(2004\)](#) is a graph-based algorithm for ranking sentences in a document based on their similarity to each other. It uses the concept of eigenvector centrality to score sentences based on their similarity to other sentences in the document. Sentences with high LexRank scores are considered to be the most important and relevant to the document. For every table i , the LEXRANK score of every $extractive_{ij}$ for table i is calculated.

Majority Voting Technique: Once the BLEU, ROUGE, and LEXRANK scores for each extractive summary $extractive_{ij}$ were obtained, we then wanted to select the most relevant and highly scored extractive summary. However, since the three metrics are different and have different ways of calculation, it was necessary to normalize the values first.

After normalizing, the majority voted summary by all the metrics was finally selected as the most relevant extractive summary as shown in the Equation (1), where Metric denotes either BLEU, ROUGE, or LEXRANK.

$$Relevant_ES = MaxVoted(MAX(Metric(abstractive_i, extractive_{ij})) \quad (3.1)$$

3.4 Model 2 – Template based system

A template-based approach is proposed next wherein we demonstrate the development of two template-based models that develop extractive summary templates representing the extractive summary of a table. These templates can be utilized by any NLP researcher to study and develop more accurate and coherent summaries and also to generate abstractive summaries.

To do this, our system first identifies a set of significant terms from each scientific paper that is downloaded. After that, an extracted summary of the tables in the article

is created using these phrases. The two systems that we have developed are explained in the subsection below.

3.4.1 TF-IDF Based System

Under this system, we have proposed two approaches namely Unigram and Bigram approach to extract templates for extractive summary.

Unigram Approach: We determine the TF-IDF score of every term in each table of each of the 1,500 papers in the corpus, eliminating stop words, non-alphanumeric characters, and extraneous punctuation marks. We also constitute a unique word list from the gold standard extractive summary of the table in concern.

We then select words that are the highest scoring terms as the template summary of that table, if and only if they are common with the unique word list of the extractive summary of that table as well. The number of terms selected is a matter of experimentation and we have considered 10,20 and 30 number of terms for our work. The evaluation is shown in Table 3.4 The Template for Match (TS) is the name given to this collection of terms. There is only one TS for all of the extractive summaries of a given table, even though each table may have many summaries.

Bigram Approach: Another approach known as the Bigram technique is devised, which takes into account two consecutive words rather than a single word. This method computes the TF-IDF score for every bigram in the document. The highest scoring bigrams, which are common in the extractive summary unique word list for that table are then selected as the template.

3.4.2 Transition Point Based System

Transition Points (TPs) are frequency numbers that divide a text's lexicon into two groups: high-frequency and low-frequency words. In this next approach we have used TP concept to show how effective it is in selecting important terms because we assume that the mid-frequency terms in the text are definitely more closely associated to a document's conceptual substance. The Transition point system uses two methods: unigram and bigram. Before applying these methods, however, we want to define some important concepts.

Let $document_i$ represent a specific document, and V_i denote its associated vocabulary, defined as the set of word-frequency pairs $\{(w_j, tf_i(w_j)) | w_j \in D_i\}$, where $tf_i(w_j) = tf_{ij}$. The term TP_i signifies the transition point within the document D_i . We can construct a set of important keywords, denoted as R_i , for that document, by selecting words that meet the following conditions:

$$R_i = \{w_j \mid ((w_j, tf_{ij}) \in V_i), (TP_i \cdot (1 - u) \leq tf_{ij} \leq TP_i \cdot (1 + u))\}$$

Here, u represents a parameter ranging between 0 and 1. Research conducted by Urbizagástegui (1999) suggests that a value of $u = 0.4$ serves as an effective threshold for this method. The transition point, TP, is calculated using the following formula:

$$TP = \frac{(-1 + \sqrt{8 \times I_i})}{2} \quad (3.2)$$

The value of I_i denotes the count of words that occur only once in the document. We can thus say that the terms whose frequencies lie in proximity to the transition point (TP) are considered significant and are given a higher weight for summarization purposes, while the remaining terms are assigned a near zero weight.

Unigram-based Approach:

In this approach, we first find out R_i , for individual papers. Next, we take the highest scored unigrams from R_i , and we consider them as the template summary for a table if these unigrams match with the unique word list created from the gold standard extractive summary of a particular table. The number of terms are a matter of experimentation. We have experimented with 10, 20 and 30 number of terms and evaluated the quality of the summary created as shown in Table 3.4.

Bigram-based Approach: In this next approach, we take R_i as our input document and calculate the TF-IDF score of each bigrams in that. We take the highest scored bigrams and choose only those which are common to the extractive summary unique word list for individual tables. This approach enabled the identification of important bigrams that provided additional context and meaning to the document as well as to the respective tables. Now the selection of terms to be included in the summary templates follow three approaches as described by López et al. (2007); Left approach, Right Approach, and Left-Right Approach.

Left Approach: Using this method, only bigrams with a TF score larger than one are chosen. The left term for each of these bigrams is chosen to be the new term added to the summary template for a particular table if the word from R_i appears in the rightmost place.

Right Approach: This approach focuses on the term which is present in right-most term of the bigram when the terms in R_i appear in the left-most position.

Left-Right Approach: Combining the first and second techniques, the final method takes into account both left and right phrases that exist in the bigram and meet the

requirement of having a minimum frequency of two.

3.4.3 Experiment and Results

This section demonstrates the manual and automatic evaluation of our proposed rule based and template based models.

Dataset Quality Evaluation - Rule based

Initially in the dataset, an abstractive summary AB_1 had multiple extractive summaries E_1, E_2 mappings denoted by $AB_i \rightarrow E_j$, where i is the total number of abstractive summaries and j is the total number of extractive summaries for each i . However, after selecting the most significant extractive summary for each table as discussed in the previous sections, we have made the dataset more relevant and compact.

We have employed two methods for validating and evaluating the quality of the corpus namely, Inter Annotator agreement-based validation and Automatic Evaluation. The following subsections provide a succinct overview of the evaluation methodology of the corpus.

Inter Annotator Agreement-based Validation: In order to validate this dataset, we employed two human annotators, A_1 and A_2 , who were tasked with evaluating the mapping between an abstractive summary and the selected extractive summary for a particular table.

Each annotator was tasked to identify whether the mappings were valid according to their opinion. A valid mapping was given a score of “1” and an invalid mapping was given a score of “0”. The dataset had 6,010 tables so the annotators were asked to validate a total of 6010 $AB_i \rightarrow E_j$ mappings.

Table 3.2 presents the confusion matrix constructed using the agreement between two annotators for both of the labels (Valid - “1” and Invalid - “0”). With the help of these scores, we then calculate the agreement between annotators A_1 and A_2 using Cohen’s Kappa¹ agreement analysis approach which is defined as follows Viera et al. (2005):

$$\kappa = \frac{Pr_a - Pr_e}{1 - Pr_e} \quad (3.3)$$

where the observed percentage of complete agreement between two annotators is denoted by Pr_a . Furthermore, Pr_e represents the percentage predicted by chance, suggesting a form of haphazard agreement among the annotators.

¹https://en.wikipedia.org/wiki/Cohen's_kappa

The final value of κ ranges from -1 to 1, with 1 denoting total agreement, -1 denoting complete disagreement, and 0 denoting agreement by chance.

The analysis of agreement using Cohen’s Kappa, in this case, shows that for the abstractive to extractive mappings, the value of κ is 0.824 with an agreement of 95% confidence interval. A higher κ value indicates a stronger agreement.

The goal of this part was to assess the effectiveness of the proposed method in accurately identifying the summary of table content in a given document.

Automatic Evaluation: To further corroborate the findings of the external annotators, we used two standard evaluation metrics: ROUGE and BLEU. BLEU determines the degree of similarity between a machine-generated summary and one or more reference summaries based on n-gram matching. On the other hand, the ROUGE family of assessment tools concentrates on recalling important information from the generated summary.

To do this, we selected all the 6010 $AB_i \rightarrow E_j$ mappings and calculated the BLEU and ROUGE scores of the extractive summary with respect to its abstractive summary. For this calculation, we used the AB_i as the reference summary and the most relevant extractive summary E_j as the candidate summary.

Table 3.3 reports the average BLEU and ROUGE-L (F1) scores for all combinations, while Figure 3.2 shows the BLEU scores that were obtained for the summary mappings for all possible combinations as mentioned above.

Similarly, Figure 3.3 depicts the Rouge scores that were obtained for the summary mappings for all possible combinations, viz. (i) *Both annotators agree*, (ii) *A1 agrees, A2 disagrees*, (iii) *A1 disagrees, A2 agrees* and (iv) *Both annotators disagree*. We have taken 40 summary mappings as its the least number of mappings in the confusion matrix above.

We may conclude from the chart analysis that the sample mappings with BLEU and ROUGE scores that both annotators agreed were VALID have higher values than the other combinations. This essentially supports our theory that the summary samples serve as the best ones when both expert annotators are in agreement, demonstrating the datasets’ quality.

No. of Mappings ($AB_i \rightarrow E_j$) : 6,010		Annotator 1	
		Valid (Score =1)	Invalid (Score=0)
Annotator 2	Valid (Score =1)	5175	40
	Invalid (Score=0)	45	210
Kappa Score		0.824	

Table 3.2: An inter annotator confusion matrix

No. of Mappings : 6,010			
Both Agree		A2 Agree	
Avg_BLEU	Avg_ROUGE-L	Avg_BLEU	Avg_ROUGE-L
94.1	0.79	49.5	0.40
A1 Agree		Both Disagree	
Avg_BLEU	Avg_ROUGE-L	Avg_BLEU	Avg_ROUGE-L
48.25	0.41	5.12	0.05

Table 3.3: Average Metrics Values

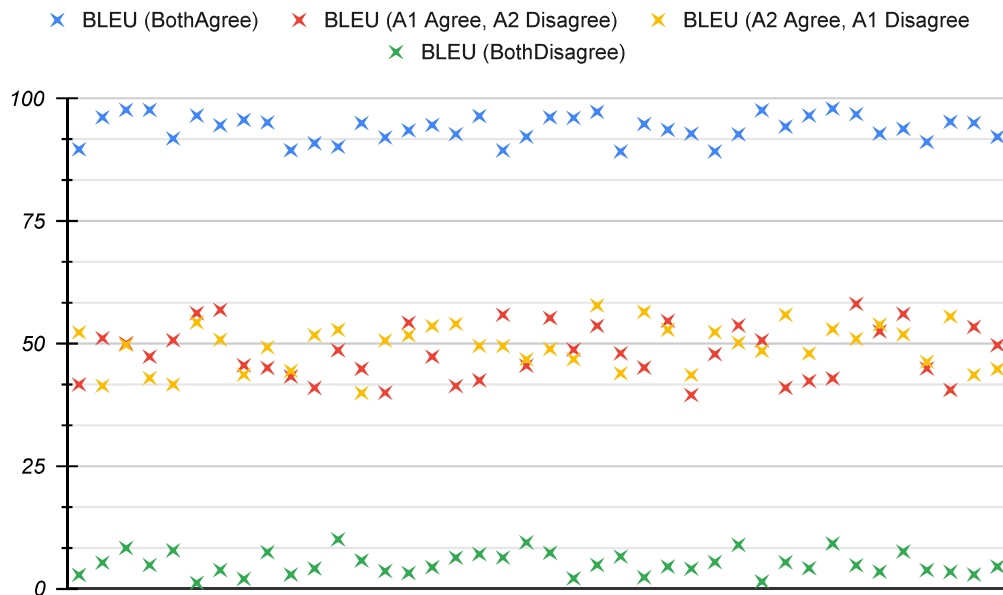


Figure 3.2: Inter-annotator BLEU scores for rule-based approach

Dataset Quality Evaluation - Template based

Automatic Evaluation: The experimentation results after applying automatic evaluation methods like BLEU and ROUGE to the output of the above-mentioned approaches are reported in Table 3.4.

After developing template-based summaries, it was noticed in the experimentation that by varying the number of terms for Template for Matching (TS), we get different BLEU and ROUGE scores for different numbers of terms. It was further noticed that BLEU scores increased with a smaller number of terms however, there was a decrease in the ROUGE score as the number of terms increase as seen in Table 3.4.

If the TF-IDF Unigram and Bigram approaches are compared, it is very clear that TF-IDF Unigram approach has better performance in extracting the summary templates.

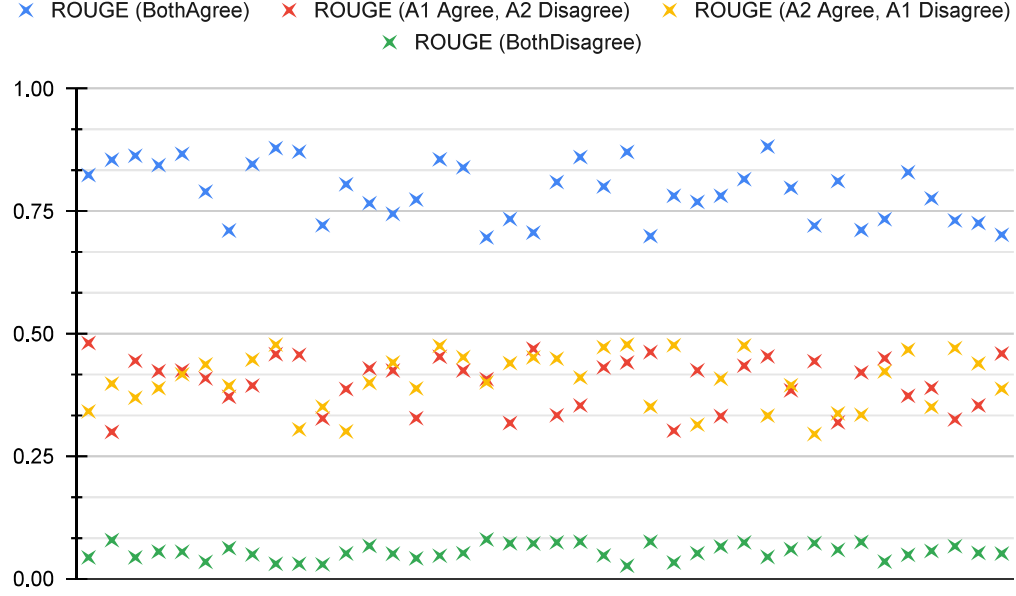


Figure 3.3: Inter-annotator ROUGE scores for rule-based approach

Moreover, if the Transition Point Approach is considered, the scores are extremely lower than the other approaches. The results of the automatic evaluation are depicted in Figures 3.4 , 3.5, 3.6 and 3.7.

If we notice Figures 3.4, 3.5, and 3.6, the observed disparity in the chart emphasizes the importance of agreement between annotators, as it directly impacts the quality and accuracy of the summaries. Thus, when both annotators reach a consensus on the validity of a summary, it can be seen that the summary in question exhibits a stronger resemblance to the summaries, as evidenced by its elevated ROUGE and BLEU scores.

Thus after using automatic evaluation measures, it can be concluded that the TF-IDF unigram approach performed better overall.

Inter Annotator Agreement-based Validation: In order to validate further, we employed two human annotators, A_1 and A_2 , who were asked to evaluate the quality of the summary templates obtained by the TF-IDF unigram approach. We selected only the TF-IDF unigram approach as it was clearly noted that it outperforms all the other approaches.

Each annotator was tasked to identify whether the mappings were valid according to their opinion. A valid mapping was given a score of “1” and an invalid mapping was given a score of “0”. The dataset had 6,010 tables so the annotators were asked to validate a total of 6010 $AB_i \rightarrow E_j$ mappings.

TF/IDF Unigram				
#Terms	METRIC(BLEU)	METRIC(ROUGE-L)		
		Precision	Recall	F-Measure
10	46	0.26	0.71	0.34
20	40	0.53	0.60	0.51
30	36	0.72	0.50	0.53
TF/IDF Bigram				
#Terms	METRIC(BLEU)	METRIC(ROUGE-L)		
		Precision	Recall	F-Measure
10	32	0.30	0.65	0.25
20	25	0.32	0.69	0.35
30	15	0.20	0.42	0.31
Transition Point System				
Bigram-Approaches	METRIC(BLEU)	METRIC(ROUGE-L)		
		Precision	Recall	F-Measure
Unigram Approach	0.04	0.14	0.17	0.08
Left Approach	0.08	0.11	0.16	0.21
Right Approach	0.11	0.19	0.21	0.22
Left-Right Approach	0.13	0.14	0.02	0.12

Table 3.4: Automatic evaluation scores for summary templates

The confusion matrix created with the two annotators' agreement scores for both labels is shown in Table 3.5.

No. of Mappings ($AB_i \rightarrow E_j$) : 6,010		Annotator 1	
		Valid (score = 1)	Invalid (score = 0)
Annotator 2	Valid (score = 1)	4,010	467
	Invalid (score = 0)	510	1,023
Kappa score		0.568	

Table 3.5: Summary template validation

With the help of these scores, we then calculate the agreement between annotators A_1 and A_2 using Cohen's Kappa agreement analysis approach. The analysis of agreement using Cohen's Kappa, in this case, shows that for the abstractive to extractive mappings, the value of κ is 0.568.

Comparison: Rule-based and Template-based Approaches

Table 3.6 shows the comparison between the rule-based approach and template-based TF-

TF-IDF Unigram Approach

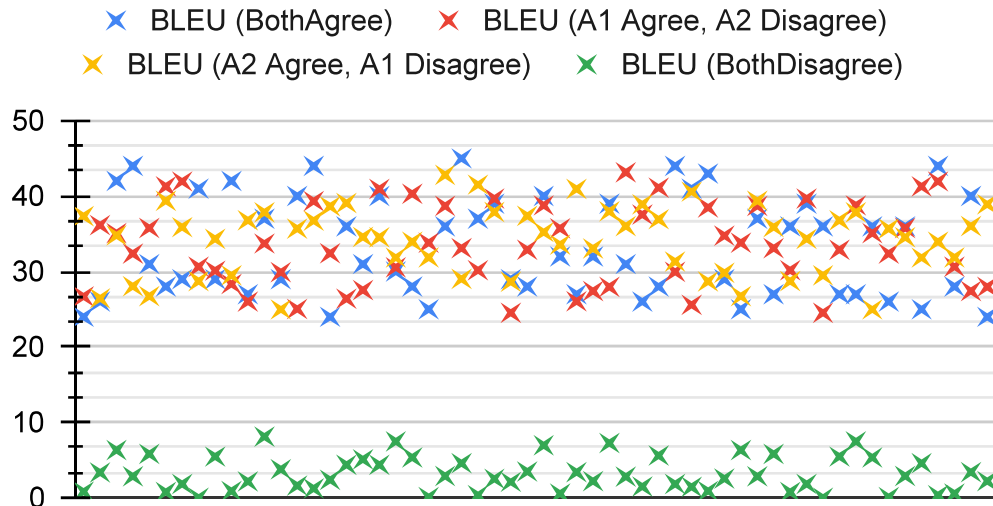


Figure 3.4: Inter-annotator BLEU scores for TF-IDF unigram Approach

IDF approach. In both cases, we have only taken those summary pairs which have been marked as valid summaries by both the external annotators. By observing the BLEU, ROUGE-L, and agreement scores we can easily conclude that the rule-based approach of dataset development is better than the template-based approaches. For this reason, we will be considering the summary generated by the rule-based approach in our next tasks.

Rule Based Summary_Both Agree		
Avg_BLEU	Avg_ROUGE-L	Kappa_Score
94.1	0.79	0.824
Template Based_TF-IDF_Unigram_Both Agree		
Avg_BLEU	Avg_ROUGE-L	Kappa_Score
48.7	0.5	0.568

Table 3.6: Comparison between rule-based and TF-IDF-Unigram approach where both annotators have agreed

3.5 Model 3 – Sense Based System

In this section , we delve into one of the key tasks within the domain of extractive summarization: the NTCIR-12 shared task. The NTCIR (NII Testbeds and Community for Information access Research) series of workshops and conferences provide a platform for

TF-IDF Unigram Approach

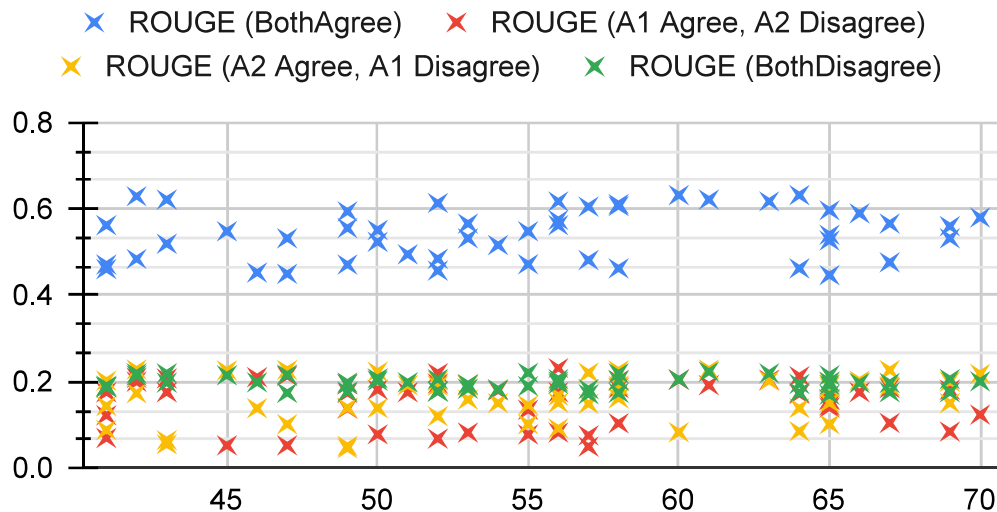


Figure 3.5: Inter-annotator ROUGE scores for TF-IDF unigram Approach

researchers to collaborate and innovate in the field of information access and retrieval. The NTCIR-12 shared task specifically focuses on two sub-tasks; ranking and summarization of web-based query outputs.

Web-based searching and information extraction present substantial obstacles, owing to the lack of a knowledge-driven classification of search queries. When users input a query, they are presented with a set of information-based links (URLs) relevant to their search. Extracting pertinent information from these links requires context analysis to identify the most relevant links and their ranking. This task is particularly challenging on mobile devices, where limited screen space necessitates efficient information retrieval methods. Thus, the primary aim of the NTCIR-12 shared task is to develop systems that facilitate the retrieval of relevant information with minimal user effort.

The NTCIR-12 shared task comprises two subtasks: ranking and summarization. In the ranking subtask, the goal is to identify and rank information units (iUnits) relevant to the query. These iUnits represent atomic chunks of information about the search query. Our method involves ranking a set of iUnits offered for certain queries based on their closeness to the query. To do this, we use a sense-based approach, presuming that the most relevant iUnits are strongly related to the query’s semantics.

In the next task based on summarization, the most significant iUnits play a crucial role in generating a two-layered extractive summary. The first layer of the summary

TF-IDF Bigram Approach

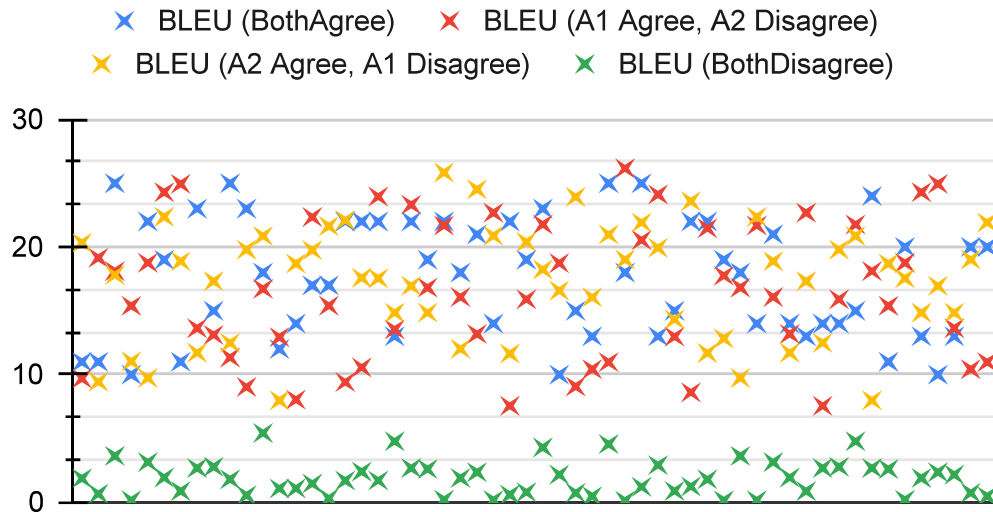


Figure 3.6: Inter-annotator BLEU scores for TF-IDF bigram Approach

consists of the most important iUnits, while the second layer connects these iUnits to provide a structured summary relevant to the query. This two-layered approach ensures that the summary remains concise yet informative. Given the nature of the task and the constraints of mobile screens, we adopt an extractive approach where each iUnit serves as an extraction unit. The system framework is shown in Figure 3.8

The next sections include an extensive overview of our proposed system and the breakdown of the test set utilised in the NTCIR-12 shared task, as well as substantial experimental findings, including evaluation measures used to assess our system’s efficiency and efficacy.

3.5.1 Experimental Dataset

For the ranking subtask, the NTCIR-12 organisers supplied both training and test data; for the summary subtask, they only had test data. Two files make up the test dataset for the ranking subtask: one file has 100 queries, and the second file has 4342 iUnits that are related to these queries. On the other hand, the summarization subtask’s test data consists of three files: intents, iUnits, and queries. The idea of intents came about in the NTCIR-12 MobileClick2 challenge. Intents are created by clustering iUnits with corresponding cluster labels, where each label represents an intent. A detailed analysis of the dataset statistics is discussed in Section 2.5.1.

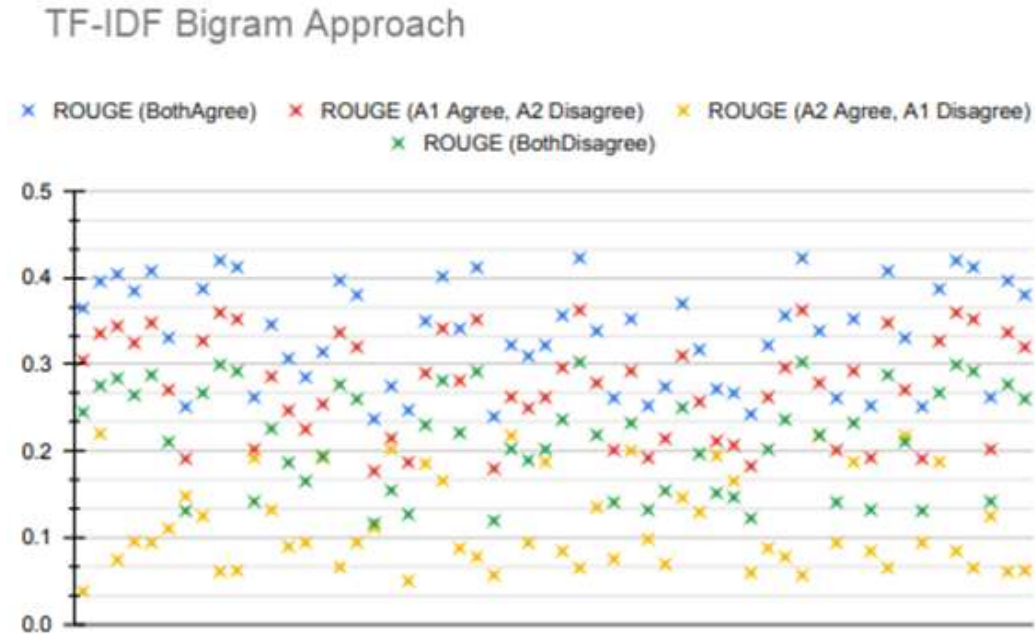


Figure 3.7: Inter-annotator ROUGE scores for TF-IDF bigram Approach

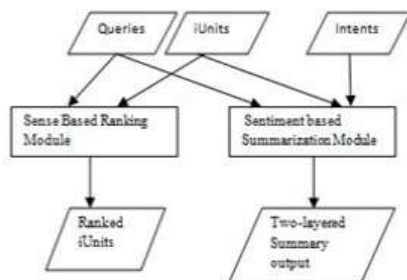


Figure 3.8: Overall System Framework

3.5.2 IUnit Summarization

The most significant iUnits and links (intents) make up the first layer of the two-layered architecture that represents the summarised output of the iUnits associated with each query. We employed the standard TextRank model along with a Wup based similarity models (Brin and Page, 1998; Wu and Palmer, 1994) to summarise iUnits given the queries.

The TextRank Model: We used a modified version of the TextRank algorithm, based on graphs, to identify key iUnits for the first summary layer Mihalcea and Tarau (2004). Text must be transformed into a graph format in order to use the TextRank model. This method involves representing ranked iUnits (words, phrases, etc.) as graph vertices and establishing links between them based on similarity. To find meaningful associations, a concept voting technique is used, in which the voting relation between vertices is shown by

the edge connecting them. Higher scoring vertices are regarded as more significant than lower scoring ones.

Here is a description of the model: Take a look at a directed graph $G = (V, E)$ that has a set of edges (E) that are subsets of $V \times V$ and a set of vertices (V). Let $\text{In}(V_i)$ represent the collection of vertices referring to a particular vertex V_i , and $\text{Out}(V_i)$ represent the set of vertices that V_i refers to.

$$S(V_i) = (1 - d) + d^* \sum_{j \in \text{In}(V_i)} \frac{1}{|\text{Out}(V_j)|} S(V_j) \quad (3.4)$$

where d is the damping factor with a value between 0 and 1. This factor incorporates the likelihood of moving from a specific vertex in the graph to a different random vertex. D is usually set to 0.85.

The iUnits have been transformed into a graph by building a similarity matrix. By highlighting the content similarities between the iUnits, this matrix generates a similarity score for each one. The following is the procedure for extracting the similarity score:

$$\text{Similarity}(S_i, S_j) = \frac{|\{w_k \mid w_k \in S_i \text{ and } w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \quad (3.5)$$

We compared the sentences represented as S_i and S_j , each containing N_i words, where $S_i = w_1^i, w_2^i, \dots, w_{N_i}^i$.

We determined the essential iUnits for the first level of the summarization task by comparing the obtained similarity score to our predetermined threshold score of 10. The output of our system is shown in Table 3.7, which shows the chosen iUnits and their associated contents for the query "The Hulk" with the ID "MC2-E-0001". The selected iUnit ID is denoted by UID in this table, and the text that goes with it is indicated by CONTENT.

The supplied intents for every query are associated with the corresponding iUnits for the second layer summary. In this connection process, the Wup-Similarity model is applied.

UID	CONTENT
MC2-E-0001-006	most famous wrestler in the world
MC2-E-0001-009	became the public face of professional wrestling
MC2-E-0001-010	made his American Wrestling Association debut.
MC2-E-0001-015	during high school, he frequently went to the Tampa Sportatorium's wrestling events.
MC2-E-0001-022	obtained his first championship in wrestling

Table 3.7: iUnit Text Corresponding to iUnit ID's

Wup Similarity Measure: Wu and Palmer (Wu and Palmer, 1994) introduced the Wup similarity model, which uses a graphical-measurement method akin to the text-rank model. The Wup similarity approach calculates the number of edges, whereas the text-rank model builds a similarity matrix through a voting process. The fundamental idea behind the Wup similarity calculation is based on calculating the distance between the graph’s root concept (R) node and concept nodes C1 and C2. The integers N1 and N2 denote the distances of the concept nodes C1 and C2 from the root node R, respectively, wherein N is the amount of distance between the closest common ancestor (CS) of C1 and C2 and the root node R. The level of commonality among two concept nodes (C1 and C2) can be calculated using the formula below:

$$WP_{\text{sim}} = \frac{2 \times N}{(N_1 + N_2)} \quad (3.6)$$

QID	IID	UID	SCORE
MC2-E-0002	MC2-E-0002-INTENT0002	MC2-E-0002-004	1.4010989011
MC2-E-0002	MC2-E-0002-INTENT0002	MC2-E-0002-010	1.29298642534
MC2-E-0002	MC2-E-0002-INTENT0002	MC2-E-0002-011	1.03626373626
MC2-E-0002	MC2-E-0002-INTENT0002	MC2-E-0002-015	1.94471802707
MC2-E-0002	MC2-E-0002-INTENT0002	MC2-E-0002-018	1.16483516484

Table 3.8: Wup-similarity scores between intents and iUnits

To calculate the context similarity score, we used the wup-similarity approach on the provided test data set’s queries and their associated intents. For each of the intents, we have determined which iUnits are most related by setting the threshold of the wup-similarity score to 1 (one). The iUnits selection method determined by a similarity threshold score of 1 is shown in Table 3.8, where QID, IID, UID, and SCORE stand for the query id, intent id, iUnit id, and wup-similarity score, respectively. The chosen iUnit ids for the question id MC2-E-0002 and intent id MC2-E-0002-INTENT0002, determined by score, are shown in the table.

3.5.3 Evaluation

For the ranking and summary portion of the NTCIR-12 Mobile-Click job (English), we have submitted two distinct run files. Here we will discuss the evaluation methods for summarization task only. The M-measure metric is used in the summary subtask to determine the system’s effectiveness.

$$M = \sum_{t \in T} P(t) \cdot U(t) \quad (3.7)$$

where T represents the set of all potential trail-texts, $P(t)$ is the probability of traversing trail t , and $U(t)$ signifies the unit-measure value of the trail.

The run we submitted for the summarising subtask achieved an M-Measure score of 11.7033. Figure 3.9 displays the average M-measure obtained for each of the four query kinds. The data shows that the M-measures for both type 2 and type 4 inquiries are low. Type 2 generated 11 flawed questions with a significantly lower M-measure score among a total of 20 queries. The inquiries with ID numbers 22, 27, 36, 37, 40, and 50 have generated a score below 3. We have noticed that the iUnits of these requests predominantly consist of numerical data such as telephone numbers, timings, etc. Out of 20 queries in type 4, 8 queries scored less than 6.

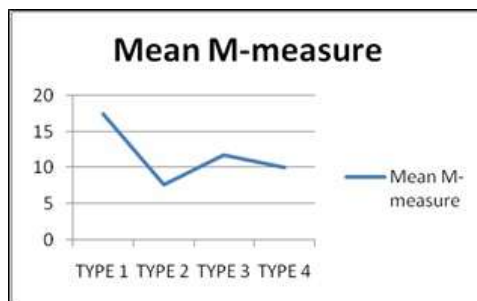


Figure 3.9: Mean M-measure for Four Query Types

3.6 Model 4 – SCISUMM: Word Vector Based System

The CL-SciSumm task consists of scientific paper summarization by treating citation phrases, known as "citances," as summaries created by the research community for cited documents. The task comprises two primary elements:

Task 1A consists of selecting specific text sections in the referenced work that accurately represent the citation. Spans can vary in size from little fragments to entire phrases or a few sequential sentences, with a limit of five sentences.

Task 1B involves identifying the specific aspect of the cited publication that each referenced text span corresponds to, selected from a predefined group.

And finally, **Task 2** involves crafting a succinct summary of the referenced article by utilising the identified cited text segments. The summary must be limited to 250 words.

3.6.1 Experimental Dataset and Preprocessing

The CL-SciSumm 2016 task used the initial training dataset from the Pilot Task TAC 2014². This event took place as part of the BiomedSumm Session during the Text Analysis Conference 2014.

However, this shared task included a new training corpus with a total of ten sets for training and a separate assessment corpus with ten different sets. Additionally, it provided three different forms of summaries for each corpus: abstracts written by research paper authors, community summaries taken from citations, and summaries produced by CL-SciSumm annotation experts. For a more comprehensive overview of the dataset, refer to Section 2.5.2.

After preprocessing, the initial reference is segmented into sentences. Three commonly used assessment methods are employed to determine the similarity among each sentence and the referenced language in the citation.

1. Jaccard's Coefficient (Trigram Model) denoted as $J(a, b)$
2. Clough and Stevenson coefficient (trigram model) denoted as $C(a, b)$
3. The Probability Measure of the Bigram Model denoted as $P(a)$

The reference sentence with the highest similarity values are termed as candidate sentences or phrases..

3.6.2 System Framework

This section includes the system proposed for tasks 1A, task 1B and task 2 in details.

Task 1A : Identification Problem

We used a straightforward method to identify the possible *Citation Texts*. The cosine similarity between each "Candidate Phrase" in the "Reference Text" and the "Cited Phrase" is calculated to get a cosine similarity value. After incrementing this score by one, the overall Cosine Similarity Score for every Candidate Sentence in the *Reference Text* is obtained.

If no *candidate sentence* scores above 1.2, then the reference text hasn't been cited. The sentence or phrase with the highest possible score is accepted as the "Citation Text" unless stated otherwise.

Task 1B : Classification Problem

²<https://tac.nist.gov/2014/index.html>

In the context of text classification, Task 1B entails the categorization of sentences found in a reference paper according to the following discourse facets: Aim, Method, Implication, Result, and Hypothesis. For this endeavour, an unsupervised approach is utilised.

The procedure is listed as following:

- Each discourse facet should be represented by its own bag of words.
- Calculate the bag vector for every bag of words.
- Determine the sentence vector for every span of cited text.
- Determine the degree of cosine similarity that exists among the sentence vector alongside every bag vector.

The discourse attribute that bears the greatest similarity to the cited text should be assigned.

Bag of Words: Every bag has a list of terms related to a particular class. Unigrams are used for creating these bags. Before building the bag, we extract the class-specific reference text from our training data. After that, we create a list of terms from these example phrases and determine each one’s tf-idf score. The words with the greatest tf-idf scores are then chosen to construct the bag.

Bag Vectors: Specifically, we used a previously trained 200-dimensional GloVe³ model trained over 2 billion tweets⁴ from Twitter to create vectors for the word bags as well as reference text.

The normalised summation of the word bag vectors that were in the previously trained Glove model’s vocabulary is used to construct the word bag vectors. Words that are not in the vocabulary are put in the null vector.

Mathematically, this process is represented as follows:

$$\vec{q}_i = \frac{1}{N_v(q_i)} \sum_{j=1}^{N_v(q_i)} \vec{W}_{ij} \quad (3.8)$$

where:

- \vec{q}_i represents the topic vector of the i th word bag.
- $N_v(q_i)$ denotes the number of words in q_i present in the vocabulary.

³(<http://nlp.stanford.edu/software/CRF-NER.html>)

⁴(<http://nlp.stanford.edu/projects/glove/>)

- \vec{W}_{ij} is the vector of the j th word in the i th word bag. If the word is out of vocabulary, \vec{W}_{ij} is set to the null vector.

Sentence vectors : The normalised summing of the vectors for each word in the sentence that was included in the GloVe model’s vocabulary is used to construct the sentence vectors. The word was placed in the null vector when it wasn’t found in the model vocabulary.

Mathematically its shown as follows:

$$\vec{t}_i = \frac{1}{N_v(t_i)} \sum_{j=1}^{N_v(t_i)} \vec{u}_{ij} \quad (3.9)$$

$$\vec{u}_{ij} = \vec{0} \quad (3.10)$$

where, \vec{t}_i is the sentence vector of the i th sentence, t_i .

$N_v(t_i)$ is the number of words in t_i present in the vocabulary.

\vec{u}_{ij} is the vector of the j th word in the i th sentence.

Cosine Similarity: The cosine similarity, S among the sentence and the topic vectors was determined using the cosine similarity measure as shown below:

$$S = \text{cosine-sim}(\vec{t}_i, \vec{q}_j) = \frac{\vec{t}_i \cdot \vec{q}_j}{\|\vec{t}_i\| \cdot \|\vec{q}_j\|} \quad (3.11)$$

The degree of similarity among the sentence vector, \vec{t}_i , and the topic vector, \vec{q}_j , grows greater when S is high, and vice versa.

Task 2: The Summarization Task: The primary aim of this task was to form a community summary, which is a structured extractive summary of the Reference Paper. From the outputs obtained from Task 1A and 1B, a community summary had to be formed, which is a structured extractive summary of the Reference Paper (RP) generated from the cited text spans of the RP.

System Design: We used Task 1b’s results as input to build the system. This includes the cited text spans with their categorised features, which, as was previously noted, consist of five types. For each reference paper (RP), the cited text spans and their corresponding facets were displayed in each Task 1b outcome.

In a reference document, a sentence’s five discourse facets are its aim, method, implication, result, and hypothesis. For every facet, stop words and duplicate entries were removed in order to improve the text spans. The cosine similarity measure was then used to calculate a similarity score between the text spans for each facet.

When the cosine similarity score was high, a random selection was made between the two sentence vectors to determine which one may be used for summary. Sentences shorter than three words were eliminated during the pre-processing stage since they did not significantly contribute to the created summary, according to the output analysis.

3.6.3 Evaluation

For Tasks 1a and 1b, we turned in two distinct runs, while for Task 2, we only turned in one. The overlap of text spans, as determined by the number of sentences in the system output relative to the gold standard, is the basis for evaluating tasks 1a and 1b. Using the ROUGE metrics, the following comparisons are used to evaluate Task 2: i) the system result with the abstract on the reference article; and ii) the system output with the gold standard summary derived from reference spans. The Task 1 and 2 outcomes for the developed system is shown in Tables 3.9 and 3.10, respectively. Examining further reveals that, for Task 1, the second runs' performance scores are typically higher. But Task 2's scores show that there's room for growth. This implies that training would probably be more effective with a larger training sample.

		Run1	Run2
Task 1a Micro Avg	Precision	0.045	0.051
	Recall	0.031	0.035
	F1	0.037	0.042
Task 1a Micro Avg	Precision	0.057	0.066
	Recall	0.037	0.046
	F1	0.045	0.054
Task 1a ROUGE2	Precision	0.058	0.058
	Recall	0.132	0.132
	F1	0.065	0.065
Task 1b Micro Avg	Precision	0.045	0.051
	Recall	0.031	0.035
	F1	0.037	0.042
Task 1b Micro Avg	Precision	0.000	0.400
	Recall	0.000	0.057
	F1	0.000	0.100

Table 3.9: Task 1 System Evaluation

		Run1
Vs. Abstract - ROUGE version 2	Precision	0.149
	Recall	0.278
	F1	0.191
Vs. Abstract - ROUGE version SU4	Precision	0.091
	Recall	0.289
	F1	0.133
Vs. Human - ROUGE version 2	Precision	0.243
	Recall	0.152
	F1	0.181
Vs. Human - ROUGE version SU4	Precision	0.249
	Recall	0.099
	F1	0.129
Vs. Community - ROUGE version 2	Precision	0.135
	Recall	0.138
	F1	0.132
Vs. Community - ROUGE version SU4	Precision	0.133
	Recall	0.138
	F1	0.119

Table 3.10: Task 2 System Evaluation

3.7 Model 5 – Embedding Based System

The goal of our forthcoming study in the extractive summary domain is to create a summary of a document by selecting pertinent and significant sentences from the corpus. In order to extract meaningful sentences, two approaches that consider both syntactic and contextual similarities between phrases have been devised. Embedding techniques like word2vec and paragraph vectors were utilised to capture semantic similarities. The outcome summary quality was evaluated using the ROUGE assessment method. We employ standard open source datasets, the specifics of which have been covered in Section 2.5, to train and test these two models.

3.7.1 Experimental Dataset

The dataset we utilised for our work is called 'Opinosis Dataset' (Ganesan et al., 2010) - Topic-related review sentences'. This dataset contains sentences collected from user reviews about a specific topic. Examples include "performance of Toyota cars" and "sound quality of bluetooth speakers", among others. There are 51 such themes in all, with each containing around 100 sentences on average. The reviews were gathered from a variety of sources, including Tripadvisor (hotels), Edmunds.com (cars), and Amazon.com etc. This dataset also includes human-generated gold summaries. The gold summaries include five human-generated summary for every document, each with a one to two-line synopsis. The details of dataset description is discussed in Section 2.5.5.

Preprocessing: Tokenization and stemming of each sentence in every document is performed using the Porter Stemmer obtained from the NLTK library. Following that, lemmatization is used to eliminate any non-ASCII characters and punctuation from each sentence. It is to be noted that lemmatization is applied to both noun and verb forms. Here are some examples that demonstrate this procedure.

- vehicles \Rightarrow vehicle \Rightarrow lemmatized (as noun)
- sleeping areas \Rightarrow sleeping area \Rightarrow lemmatized (as noun)
- exists \Rightarrow exist \Rightarrow lemmatized (as verb)
- informing \Rightarrow inform \Rightarrow stemmed

3.7.2 Word Embeddings with Weighted Mean Vectors

This method is extremely useful for analysing and learning word embeddings from raw text. It consists of two categories: the Continuous Bag of Words (CBOW) model and the Skip-Gram model. The two approaches are similar, but with one important distinction: CBOW predicts target words from context, whereas Skip-Gram predicts context from target words. We offer a novel supervised technique for constructing sentence vectors that is fairly simple.

Algorithm : Sentence Embedding

Input: Word embeddings v_w where $w \in V$, a set of sentences S , parameter a , and estimated probabilities $p(w)$ where $w \in V$ of the words.

Output: Sentence embeddings v_s where $s \in S$

1. **for all** sentence s in S do

2. $v_s \leftarrow \frac{1}{s} \sum_{w \in s} \frac{a}{a+p(w)} v_w$
3. **end for**
4. Compute the first principal component u of $v_s - s \in S$
5. **for all** sentence s in S **do**
6. $v_s \leftarrow v_s - uu^T v_s$
7. **end for**

Paragraph Vector Model

This section describes CBOW⁵, how it is trained, and how paragraph vector is simply an extension of it.

$$h = \frac{(w_1 + w_2 + \dots + w_c)}{c} \quad (3.12)$$

The update procedures stay the same, with the exception that each word in context C must have the update applied in order to modify the input vectors.

$$W_{w_I}^{(t)} = W_{w_I}^{(t-1)} - \frac{1}{C} \cdot \nu \cdot EH \quad (3.13)$$

The model's goal is to maximise the average log probability provided with a sequence of training words.

$$\frac{1}{T} \sum_{t=k}^{T-k} \log p(w_t \mid w_{t-k}, \dots, w_{t+k}) \quad (3.14)$$

A multiclass classifier softmax does the prediction. There, we have:

$$p(w_t \mid w_{t-k}, \dots, w_{t+k}) = \frac{\exp(y_{w_t})}{\sum_i \exp(y_i)} \quad (3.15)$$

y_i is the non-normalized log-probability for the output words i , which are computed as

$$y = b + Uh(w_{t-k}, \dots, w_{t+k}; W) \quad (3.16)$$

where U , b are the softmax parameters. h is constructed by a concatenating or averaging word vectors which are extracted from W . Using text fragments of varying lengths, the Paragraph Vector model seeks to develop continuous representations of fixed length. These representations find applicability in a variety of natural language processing problems by fusing word semantics with bag-of-words features. A small addition to the Word2Vec concept is Paragraph Vector. It predicts the following word in a sentence, just

⁵Continuous bag of words

like Word2Vec. Nevertheless, the way h is computed in Paragraph Vector is not the same as in the original model. Paragraph vectors are represented by a new matrix D in addition to the matrix W . Every paragraph is handled as a separate word in this paradigm. But unlike words, which have a common vector representation across contexts, paragraphs have distinct vector representations. Averaging a paragraph vector d with a context of word vectors C at each step yields h :

$$h = \frac{(D_d + w_1 + w_2 + \dots + w_c)}{C} \quad (3.17)$$

With the exception of updating the paragraph vectors, the weight update functions are the same as in Word2Vec.

The paragraph and word vectors are trained using stochastic gradient descent. The backpropagation method is used to obtain the gradient. There are two distinct stages: (1) training to obtain the softmax weights U , b , and word token vectors W ; and (2) obtaining D , the paragraph vector representations (of the previously viewed paragraphs). 2) After adding more columns to D and gradient descending on D , the paragraph vectors of the previously unseen paragraphs stay stable, but the values of W , U , and b change. Predictions for particular labels are made using logistic regression.

3.7.3 System Framework

Pagerank and Networkx: We employed the well-known sentence ranking method PageRank (Page et al., 1998) to rank sentences within each document. We saw each sentence ID in the document as a network node and used PageRank to retrieve the sentences that were ranked. Specifically, we used the Networkx module (Hagberg et al., 2008) to construct page ranking by setting the power method eigenvalue solver’s maximum iteration parameter to 200. Metadata formats for graphs, graph algorithms, generators, and drawing tools are provided by Networkx.

The PageRank algorithm involves the following computation of a weight value between two nodes, which indicates how similar the nodes representing phrases are to one another:

$$\begin{aligned} \text{similarity}(s_i, s_j) = & \text{correlation}(\text{vec}(s_i), \text{vec}(s_j)) \\ & + \text{overlap_tokens}(\text{lemmatized_stem}(s_i), \text{lemmatized_stem}(s_j)) \\ & + \text{overlap_postags}(s_i, s_j) \end{aligned} \quad (3.18)$$

In this instance, the similarity between sentences i and j is represented by $\text{sim}(s_i, s_j)$.

The nodes that represent the two sentences are weighed differently based on this similarity value.

We normalised the correlation values to fall between 0 and 1 in order to ensure compatibility with the potential range of -1 to 1 for correlation values and to solve the restrictions of the PageRank module in handling negative weights. We also included other characteristics, such the quantity of tokens that were shared by the lemmatized and stemmed versions of sentences I and J. In order to calculate sentence similarity, we also took into account sentences that had the same part-of-speech tags.

Furthermore, we used correlation as a similarity metric between sentence vector representations, acknowledging that certain sentences can have little token overlap and yet stay similar, or on the other hand, have token overlap yet differ because of negation situations.

Word2vec, Glove and Paragraph vector: We use word2vec and doc2vec as two separate modules for sentence vector generation. 3.7.2 contains an earlier discussion of the algorithms used for this purpose.

The predicted frequency $p(w)$ of each term is obtained from datasets including enwiki, poliblogs, text8 (Arora et al., 2017), and commoncrawl, as explained in the algorithm. For our work, we set the parameter "a" to a constant value of 3×10^{-3} . The vectors of words that are absent from the collection are initialised at random. The value of "a" can be changed to fine-tune the result. This is the algorithm that our first model uses to generate sentences.

We also use paragraph vectors for sentence vector generation, as described in 3.7.2, in addition to word2vec. For this, we make use of a trained doc2vec model.

We train on two corpora: the first is WIKI, which is the full English Wikipedia; the second is AP-NEWS, which is a collection of news stories from 2009 to 2015.

Based on the previously elaborated principles and the aforementioned section, we have implemented three models.

Model 1: Glove + Word overlap + Pos Tagging: For sentence vector creation to be fed to the pagerank module, the first model makes use of the integrated GloVe vectors that were trained on the Gigaword and Wikipedia2014 corpus. The calculated word frequency is then used to weighted average these word vectors, as was previously said. The smoothing invariance factor technique 3.7.2 is then used to construct sentence vectors.

Model 2 : Word2vec + Word overlapping + Pos Tagging: Model 2 shares similarities to Model 1, but it uses Google News Word2vec instead of GloVe vectors. Moreover,

POS tag overlap between phrases is taken into account while determining sentence similarity.

Model 3 : Paragraph + Word overlap + Pos Tagging In contrast to Models 1 and 2, Model 3 does not take the weighted average of word vectors into account when generating sentences. Rather, it uses the paragraph vectors in 3.7.2 to construct word vectors. As a result, in Model 3, normalised values of word overlap, POS tag overlap, and correlation similarity between the sentence vectors are used to compute similarity scores across sentences.

3.7.4 Evaluation

We use the following procedure to assess our model. First, every human-generated summary for every document is combined. The Rouge score between our summary and the combined summary is then determined and the results are shown in Table 3.11. The following co-occurrence statistics describes the assessment measure Rouge (Lin, 2004) which we have used:

$$ROUGE - N = \frac{S \in ReferenceSummaries Count_{match}(gram_n)}{S \in ReferenceSummaries Count(gram_n)}$$

where n is the length of the n -gram, $gram$, and $Count_{match}(gram_n)$ is the maximum number of n -grams common in a system summary and a reference summary.

Model Number	Model Description	Rogue - 2 Score
1	Glove with Word Overlap and POS Tagging	24.0%
2	Word2vec with Word Overlapping and POS Tagging	25.9%
3	Paragraph Vector with Word Overlap and POS Tagging	26.7%

Table 3.11: Rogue - 2 scores

3.7.5 Discussion

It can be observed that the proposed models effectively identify the most important sentences inside a given manuscript. Notably, these models integrate structural and semantic similarities in addition to simple syntactic and lexical comparisons between sentences. Furthermore, when taking into account token overlap based on lemmatized or stemmed tokens within phrases, they do better in terms of rogue scores. The quality of sentence vectors is highly dependent on the parameter "a" in equation 3.7.2, hence it is necessary to assess model performance at different "a" values.

In fact, the outcome shows that our paragraph vectors work fairly effectively. One benefit of utilising paragraph vectors may be that they may be trained on unlabeled data, which makes them useful for situations when there is a lack of labelled data. In addition to tracking word semantics, paragraph vectors also record a sentence’s word order.

As future work, clustering is another technique for extractive summarization that is worthwhile to take into account. We can apply PageRank within each cluster to group sentence vectors produced by either model and choose representative phrases.

3.8 Observations

Throughout this chapter, we conducted an in-depth examination of numerous extractive summarization approaches, presenting our novel contributions targeted at furthering this dynamic field. Our research began with a thorough focus on summarising tabular data taken from scientific publications. We presented two separate approaches: a rule-based strategy and a template-based approach. We improved our dataset’s quality by rigorous review, which included dataset quality assessment and validation procedures. This refinement method improved the dataset’s relevance and compactness, allowing for more efficient information extraction from tables.

Moving on to the NTCIR shared goal, we created a sense-based system for efficiently ranking and summarising English inquiries. Using sentiment lexicons and tabulation-based techniques, our system achieved encouraging results, demonstrating its ability to minimise user involvement with mobile devices while retrieving relevant data.

Furthermore, our research expanded to summarising scientific papers, as evidenced by our participation in the SCISUMM 2017 shared assignment. We designed tools to extract crucial information from scientific articles, drawing on our expertise in extractive summarization. These efforts helped to streamline information retrieval and knowledge transmission in academic communities.

Next, we investigated the use of word embedding approaches into extractive summarization. We wanted to improve the coherence and informativeness of extracted summaries by using semantic representations stored within word vectors. Our experiments and evaluations yielded detailed insights into the efficacy of various approaches, offering light on how they affect summary quality.

We used rigorous approaches to evaluate our systems, including inter-annotator agreement-based validation and automated evaluation measures like BLEU and ROUGE. These assessments provided vital insights into our approaches’ performance and efficacy, allowing

us to make educated decisions for future refinement and enhancement.

Overall, our contributions to this chapter demonstrate our commitment to developing the discipline of extractive summarization. By addressing multiple issues across numerous datasets and activities, we hope to promote effective information extraction and knowledge representation, ultimately contributing to the broader landscape of natural language processing and information retrieval.

Chapter 4

Abstractive Summarization

4.1 Introduction

In today’s age of copious digital information, compressing substantial knowledge into succinct but informative summaries is critical. This problem is solved by automatic text summarization, which creates a concise, logical synopsis while preserving important details and conveying the main ideas of the source material. As emphasised by [Radev et al. \(2002\)](#), a summary should ensure manageability and brevity while successfully conveying important elements from the original material.

Automatic text summarization can be achieved through two main approaches: extractive and abstractive. Conventional methods of extractive summarization entail choosing and rearranging preexisting phrases or sentences from the original material. These techniques, meanwhile, don’t always succeed in preserving the original content’s rich context and significance. Conversely, abstractive summary techniques, as explained by [Allahyari et al. \(2017\)](#), aim to get around these restrictions by creating summaries that retain the information’s coherence and essential elements while also condensing it. Since it necessitates real-world data and semantic class analysis, abstractive summarization is thought to be more difficult than extractive summarization ([Sunitha et al., 2016](#); [Fabbri et al., 2019](#)).

It is however true that the abstractive method is considered better than the extractive method, even if it is more complex. This is because abstractive approaches produce a summary that is more meaningful than human-generated summaries since they closely resemble them ([Al-Radaideh and Bataineh, 2018](#)). Acceptable results for both kinds of summaries should contain sentences that, with the least amount of repetition, retain the primary ideas and concepts found in the original text. In order to maintain the sense

of the text, even with lengthy phrases, sentences should also be cohesive and consistent (Sunitha et al., 2016). Moreover, the final summary must also successfully convey the most important details from the original material in a succinct manner.

An extensive understanding of abstractive summarization requires a review of foundational studies and important publications in the field. To address the challenges of producing abstractive summaries, academics have tried a range of approaches and techniques over time (Raphal et al., 2018). Prior methods focused mostly on creating linguistic patterns and rule-based processes to transform incoming data into concise summaries. But the emergence of deep learning (Suleiman and Awajan, 2020) and neural network architectures changed the face of abstractive summarization and made it possible to develop increasingly complex and data-driven techniques. Significant advancements in automated summarising tasks have been made possible by the use of sequence-to-sequence models, attention mechanisms, and transformer designs, which have significantly improved the quality and fluency of generated summaries (Zhang et al., 2018).

Building on the foundation created by earlier studies, this chapter presents our significant contributions in the field of abstractive summarization.

Our research is centred around deep learning techniques, particularly the RNN approach. The models undergo thorough training and testing on reputable open-source datasets like CNN/Daily Mails and DUC, guaranteeing the validity and applicability of our results. Our goal is to use recurrent neural networks to generate summaries that are both coherent and informative, capturing the essence of the original content while also capturing the semantic nuances of the input text.

In addition, we present a novel approach that goes beyond conventional summary methods by using one-word abstractive summaries that are taken from our own carefully created performance analysis dataset. This innovative method has great potential for condensing complex numerical data on student performance into concise and useful insights. Our goal is to enable educated decision-making and intervention techniques by providing educators and stakeholders with this important information, which will ultimately lead to beneficial outcomes in educational environments.

After developing models, we also conducted comprehensive tests on a variety of datasets covering different areas in order to assess the efficacy and resilience of our proposed solutions. The experimental framework includes metrics for model training, dataset selection, and assessment that are customised to meet the unique goals of each methodology. Our goal is to show, via thorough testing and analysis, how effective our suggested ap-

proaches are in producing abstractive summaries in different contexts. We contribute to the current progress of abstractive summarization research by presenting our suggested systems and experimental results, setting the foundation for next advancements and applications in this quickly developing subject.

The chapter’s remaining sections are arranged as follows: Section 4.2 outlines the state of the art literature in the abstractive summarization domain, Section 4.3 discusses our proposed deep learning based model, Section 4.4 then outlines our proposed performance analysis system and finally Section 4.5 describes the conclusion of the chapter.

4.2 Survey

The goal of abstractive summarization, is to understand and paraphrase the input text in order to produce succinct and coherent summaries. Abstractive summarization approaches have come a long way in the last few years, mainly due to developments in neural network topologies and deep learning models. The fluency and coherence of traditional methods of abstractive summarization were frequently lacking. However, the quality of abstractive summaries has significantly improved as a result of recent developments in deep learning and neural network architectures. This overview examines these recent advancements, emphasising important approaches, difficulties, and potential paths forward for the discipline

Abstractive summary has been revolutionised by models based on transformers, like BERT (Devlin and et al., 2023) and generative GPT (Brown and et al., 2020). These models make use of self-attention mechanisms to produce summaries that are rich in context and capture long-range interdependence. Different versions of these models, such as T5 (Raffel and et al., 2019) and BART (Lewis and et al., 2019), have been developed specially tuned for summarization tasks and have achieved outstanding results on several benchmarks. Liu and Lapata (2020) presented PreSumm, a transformer-based architecture-integrated neural model for abstractive text summarization that produces novel outcomes on a range of summarization datasets.

Using reinforcement learning (RL) to maximise summary creation is another new trend in abstractive summarization. To reduce exposure bias and increase fluency, models such as networks working with pointer-generator (See et al., 2017) integrate supervised learning with reinforcement learning. RL-based methods incentivize models to produce logical and informative summaries, resulting in outputs that are more akin to those produced by humans. (Pasunuru and Bansal, 2017) focused on generating video captions but leveraged entailment rewards to reinforce abstractive summarization, improving the quality and

informativeness of the generated captions. (Paulus et al., 2018) in 2018, introduced a deep reinforcement model designed for abstractive summarization which produces high quality summaries. Ma and Sakti (2020) explored the integration of external knowledge sources into pre-trained transformers for abstractive summarization, aiming to enhance the summary generation process with additional information.

Abstractive summarization has widely embraced encoder-decoder architectures, made popular by sequence to sequence models (Sutskever and et al., 2014). The aforementioned models consist of an encoder that processes the text input and a decoder that generates the summary. Versions of encoder-decoder models, including the Transformer architecture put out by (Vaswani et al., 2017), have further improved their efficacy in summarization tasks. Using seq-to-seq RNNs, Nallapati et al. (2016) presented an abstractive text summary strategy and investigated methods to enhance the level of quality of output summaries. Liu et al. (2019) studied how using BERT, a pre-trained encoder, for text summarization, achieves improved performance by leveraging the rich contextual representations.

Some other popular works have also been done in this field. Gehrmann et al. (2018) proposed a bottom-up abstractive summarization approach that incrementally constructs a summary by predicting key content selection and generating natural language phrases. Zhou et al. (2018) outlines a neural document summary system that efficiently combines extractive and abstractive approaches to provide clear and detailed summaries. The algorithm jointly learns to score and pick phrases. See et al. (2017) in their paper, explored a network-based pointer generator for summarization, which combines extractive and abstractive methods to generate summaries by copying content from the original data. Li et al. Li et al. (2021) proposed DRGAT, a dual-reading graph attention network, which leverages graph attention mechanisms to capture global and local dependencies for abstractive summarization.

There are still a number of obstacles in the way of abstractive summarization, despite tremendous advancements. More research is needed in the areas of processing out-of-domain inputs, factual correctness and hallucinations, and producing coherent and informative summaries across several domains. Furthermore, objectively assessing abstractive summaries' quality is still a difficult task. Incorporating outside knowledge, improving model interpretability, and investigating controllable generation mechanisms are some potential future avenues in this discipline.

4.3 Model 1 - Deep Neural Network Based System

Deep neural networks are capable of understanding and generating temporal sequences of data. They have shown amazing performance in areas like computer vision, machine translation, and speech recognition. While deep neural networks are typically constructed as feedforward networks, recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM), have demonstrated extraordinary efficacy in applications such as language modelling.

RNNs are particularly useful with sequential data since each unit may utilise its own internal storage to remember information from previous inputs. This skill is crucial for interpreting linguistic nuances, such the difference between **"I have washed my hair"** and **"I had my hair washed."**. Recurrent neural networks (RNNs) have the potential to anticipate words at the conclusion of sentences with greater accuracy since they can take context into account right from the beginning. However, vanishing gradients is a common problem with RNNs. Consequently, this motivated us to use Long Short-Term Memory (LSTM) networks rather than RNNs. Longer sequences can be better retained by LSTMs since they can hold onto some of the prior context.

The implementation of abstractive summarization using the encoder-decoder RNN, will be covered in the following parts. This idea is inspired by the efficacy of (Bahdanau et al., 2014) neural machine translation models. It employs an attention-based encoder that learns a soft alignment across the input text in order to help create the summary.

4.3.1 RNN encoder decoder

Recurrent neural networks operate over a variable-length sequence $x = (x_1, x_2, x_3, \dots, x_T)$. They are composed of an optional output o and a hidden state, indicated as s_t . Using the previous hidden state s_{t-1} and the current input x_t , the hidden state s_t is updated by a function f . f is a function whose complexity can change. This framework's main goal is to convert a variable-length sequence into a vector format with a fixed length. It can also decode a vector representation with a fixed length back into a sequence with a variable size.

The encoder, an RNN, modifies the hidden state in accordance with the sequential processing of input symbols. On the other hand, the output sequence is produced by the decoder, which is also an RNN, by forecasting the next symbol o_t using the hidden state s_t . The hidden state of the decoder at time t is computed by taking into account the hidden state that existed before, s_{t-1} , the symbol that was previously created, y_{t-1} , and

maybe other contextual data, h .

Neural attention model: The decoder in the encoder-decoder architecture is responsible for producing text based only on the encoder’s final hidden state, C . It is anticipated that this vector will contain all pertinent information regarding the source text. But it is unrealistic to expect the decoder to generate reliable translations based only on a single vector that may contain lengthy words. Researchers have found that reversing the source sequence produces noticeably superior outcomes even when LSTMs are used. In a similar vein, giving the input twice helps the network memorise it better. Rather than trying to encode the complete source sentence into a fixed-length vector, an attention method is incorporated. Rather, at each stage of the output creation process, the decoder is free to concentrate on distinct portions of the source text.

In order to assist the decoder in selecting only the coded inputs that are crucial for each stage of the decoding process, the attention model stands between the encoder and the decoder.

Sampled softmax and Output projection: Let us assume that the existing problem is a single-label problem. This means that every training sample $(x_i, \{t_i\})$ consists of one target class and one context. $P(y | x)$ represents the likelihood of the target class being y , in the given context x . Our objective is to train a function such as $F(x, y)$ to produce softmax logits, which, given the context, indicate the relative log probability of the class:

$$F(x, y) = \log(P(y | x)) + K(x) \quad (4.1)$$

In this case, $K(x)$ stands for any function that is independent of y . Typically, in full softmax training, we would compute logits $F(x_i, y)$ for all classes y in the universe L for each training sample (x_i, y_i) . However, in the case where the universe of classes L is enormous, this procedure may become computationally costly.

In the concept of “Sampled Softmax”¹, for every training example (x_i, t_i) , we selectively pick a small set $S_i \subset L$ of “sampled” classes. This selection is made according to a designated sampling function $Q(y | x)$.

4.3.2 Experiments and Discussion

Dataset Description: We used the CNN and Daily Mail newspaper databases for our analysis. There are more than 300,000 documents in this dataset. Every document has

¹https://www.tensorflow.org/extras/candidate_sampling.pdf

Parameters	Description	Value
seq_length_x	The encoder side sequence length	41
seq_length_y	The decoder side sequence length	16
vocabulary size	Total number of words for first sentence of article + Total number of words for headlines + num + unk + pad + eos + bos	86854
embedding dimension	Vector dimension for every word	512
depth	No. of layers on both the encoder and decoder sides	3
memory dimension	Hdden vector dimation of cells in LSTM	512
keep_prob	Dropout parameter for training	0.5
# samples	These are samples used for sampled softmax loss computing	5
rate of learning	The updating process affects the weights' current value.	0.001
optimizer	Algorithm used for Optimization	Adam
loss function	Sequence loss - average log-perplexity per symbol	-

Table 4.1: Parameters used in the seq to seq Model

three to five news headlines that match to each news article.

From this corpus, we retrieved 583474 news and headline pairings. We deleted all grammar, replaced all numbers with "num" tokens, and lowercased all words. The details of the dataset is discussed in more detail in Section 2.5.4. The parameters of the seq to seq model used are described in Table 4.1

Sentence length selection: We used the CNN and Daily Mail newspaper databases for our analysis. There are more than 300,000 documents in this dataset. In order to strike a balance between the amount of training samples and sample length, the minimum and maximum lengths of the article and headline have to be determined. A well chosen sentence length reduces the amount of padded sequences in each sentence, which lessens the bias in the model towards padded sequences. The article's first sentence's minimum and maximum lengths in the present work are set at 13 and 41, respectively. The headlines'

minimum and maximum lengths, which are 7 and 16, respectively, have been determined. Previous sections implied that a consistent length had to be given for the inputs that the labels and the encoder needed to receive. Therefore, the maximum lengths of 16 and 41 for the label (headline) and input (article), respectively, have been chosen. All sentences that are part of an article are either shortened if they are longer than 41 or padded if they are shorter than 41. In a similar vein, all sentences intended for headlines are padded or reduced. Almost 430631 article and headline pairs were obtained after doing this.

Vocabulary formation: It is difficult to change the vocabulary size. It is recommended that the proportion of new terms in the vocabulary should not be higher than 5%. Balance is additionally required in the sequence’s duration and the amount of padding tokens . In our dataset, we found 118,649 distinct terms. After removing terms that just once appeared, our vocabulary amounted to 86,849 terms. We have also added special tokens for new words, the start of a sentence (bos), and the conclusion of a sentence (eos). A distinct token ” ” is used to represent each number token. We’ve included a few samples of data from my training dataset here:

Data sample 1: “With around $< num >$, individuals evacuated from fire-ravaged districts, Russian authorities reported on Sunday that firemen had taken control over blazes spreading across tens of thousands of acres in western Russia.”

headline: “Approximately 50% of the several wildfires in Russia have been doused or brought under control..”

Data sample 2: “Ashley Cole has joined the prestigious Italian club Roma after being released by Chelsea at the close of the previous season.”

headline: “Cole has signed a contract with Serie A team Roma.”

Discussion: It is clear that the sequence-to-sequence architecture is a complex structure that is taught using an article’s headline and first phrase. The main goal is to retain the semantic core of a longer input text while condensing it into a shorter sentence. It can be seen that this suggested model has a number of fascinating characteristics. The first stage is the input sentence, and the headline is created from the output sentence. These lengths were chosen with care to reduce the amount of padding symbols in our dataset and to guarantee that important information is kept intact even after truncation. The model might show bias in favour of these unknown elements if our vocabulary contains an

excessive number of unknown tokens.

Because our model’s vocabulary size was too large, we computed the loss using sampled softmax in order to speed up training. On our test set, the rogue score that was calculated was not very good. Through examination of the test set’s computed output, we discovered that the model generated a significant amount of pad and unknown token.

4.4 Model 2 - Performance Analysis Based System

The advancement of educational technology in diverse applications has been greatly facilitated by the integration of Natural Language Processing (NLP) techniques in recent decades. These include of conversational technology for tutoring, automated scoring systems, and the development of customised learning resources. Furthermore, current research projects are concentrated on creating a student performance analysis system in the education sector.

Such a system is required because standard topic assessments are not able to sufficiently evaluate students’ talents. It is imperative to conduct a thorough assessment that include year-round evaluation of non-scholastic disciplines including physical education and personal attributes. But creating an automated system like this is difficult, especially in rural locations where there are problems with data availability and low instructor involvement. Inspired by these difficulties, we have attempted to create a performance analysis system that makes use of natural language processing (NLP) methods to help parents and educators assess the general strengths and weaknesses of their kids.

We propose a system for generating one-word abstractive summaries in order to assess student performance thoroughly. It’s possible that traditional subject-based tests don’t fairly represent a student’s total performance. Medical emergencies are one example of a factor that can impact individual subject scores, which makes overall assessment difficult. It is also difficult to analyse performance across multiple subjects. Our objective is to create an automated system that uses one word abstractive summaries to classify subjects and evaluate the strengths and weaknesses of students. The subject class identification and overall performance analysis modules make up this system.

Preparing the Experimental Dataset: We gathered information from five educational institutions and preprocessed it to produce an experimental dataset with 2045 student outcome samples from a range of subjects. A team of educators then verified and annotated this dataset.

Subject-Class Mapping: In order to overcome this difficulty, subject-matter experts assigned subjects to classes in accordance with National Board of Accreditation (NBA) Programme Outcome (PO) recommendations. Soft skills, domain knowledge, programming skills, logical and quantitative aptitude, and advanced knowledge for engineering disciplines are the five suggested courses.

Overall Performance Summary: Our system attempts to provide us with an abstractive summary output by identifying strengths and weaknesses based on subject-specific courses and overall performance.

System Validation: The subject classes and overall performance subsystems were used to validate the proposed system. Both were verified with a test dataset that is a subset of the experimental dataset and the previously indicated supervised classifiers.

4.4.1 Experimental Dataset

A precise, organised dataset is needed to build a performance analysis system. The structured data required to build the system is not available from any of the earlier studies conducted in this area. Thus, an unstructured dataset has been obtained from five distinct educational organisations. Moreover, we also wanted to protect the privacy of the student data like name and phone number which are private in nature. For this reason, anonymous 2564 student result samples are included in the gathered dataset. To annotate the dataset, certain number of teachers labelled the experimental dataset using two labels: the first was for five distinct subject classes, and the second was for the overall performance of the student, including their strengths and weaknesses. The dataset development has been discussed in detail in section 2.4.

4.4.2 Identification of Subject Classes/Categories

For the 50 distinct engineering subjects that are included in our experimental dataset, we suggest five distinct classes: domain understanding, skills in programming, interpersonal skills, quantitative and logical abilities, and advanced knowledge. For a student, these sessions aid in understanding the effectiveness of a certain set of disciplines. Additionally, it offers a sufficient output to evaluate a student's overall performance in comparison to subject-specific assessments.

A team of educators have organised subjects to five distinct classes by utilising their domain expertise and adhering to NBA-based programme targets. Furthermore, the courses are made to satisfy the needs of the business world as well as academia. For instance, the soft skill course covers topics like group discussions, economics for engineers, ethics in the workplace, speech and report writing, and English language. An engineering program's subject distribution in classes is depicted in Figure 4.1.

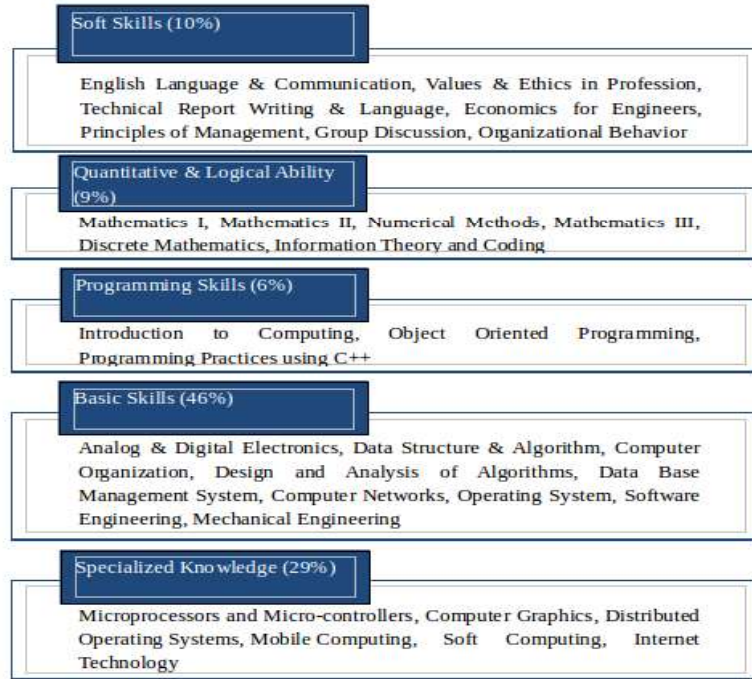


Figure 4.1: An outline of the recommended classes and the subjects they would cover.

We trained two popular classifiers on our experimental dataset, Naïve Bayes and Logistic Regression, to create an automatic subject class identification algorithm. To construct the modules, we extracted out a number of parameters from the training dataset, such as 10th and 12th grade results, marks for semester and midterm exams, total marks, DGPA, etc.

Because the feature set was unique, Naïve Bayes was helpful in class assignment, and Logistic Regression produced good results by providing confidence scores for class assignment. The system was built following these steps:

1. Create a feature vector, F_V , by extracting different features from the training dataset. Tuples representing different features make up F_V . Examples of these are (1, 0, 1, 1, 1, 1), (1, 1, 0, 1, 0, 1), (1, 1, 0, 1, 0, 0),..., (1, 1, 1, 1, 0, 1).

2. The feature vector should have five labels, representing the classes of the subjects: programming abilities, domain expertise, advanced knowledge, soft skills, and quantitative and logical ability.
3. Build the two methods, M_{NB} and M_{LR} , respectively, by applying the feature vector and its accompanying labelling through Naïve-Bayes and Logistic Regression classifiers.
4. For the proposed approach, combine these methods using the union operator to create a new technique, $M_{Combined}$.

4.4.3 Strong-Weak Classification

Following the identification of topic classes, we suggest an additional model that use a one-word abstractive summary method to evaluate students' overall performance. Using a team of teachers with specialised knowledge in the subject, this approach assigns students to strength and weakness classes and provides detailed feedback. To create the system, we use both feature extraction techniques and machine learning classifiers. For constructing the performance analysis system, we first extract different elements from the training dataset, such as subject scores and recognised topic classes. The final abstractive summary for each student is then produced by integrating the results of the Naïve Bayes and Logistic Regression exercises. The next section describes how the output of both systems in the experimental dataset is verified by the test dataset.

4.4.4 Experiments and Results

We employed test datasets with both classifiers present in order to assess the suggested systems. To further demonstrate the significance of specific features we used in developing the performance analysis system, we have also carried out an ablation study. F-measure, recall, and precision are the formats in which the ultimate results and the ablation result have been reported.

Validation of Subject Classes: In order to validate the output obtained from the system we proposed, labelled test data have been used which is discussed in Section 2.4. We evaluated the test dataset utilising three classifiers— $M_{Combined}$, M_{NB} , and M_{LR} —in order to ascertain the quality of subject classification. Recall, precision, and F-measure calculations have been examined in order to get a comparison of the classifier performances as shown in Table 4.3.

Classes		Model 1(Naïve Bayes)			Model 2(Logistic Regression)			Ensemble of Model 1 and 2		
		Precision	Recall	F measure	Precision	Recall	F measure	Precision	Recall	F-measure
Soft/Interpersonal Skills	A	0.73	0.69	0.71	0.75	0.63	0.68	0.71	0.75	0.73
	B	0.68	0.70	0.69	0.70	0.63	0.66	0.80	0.73	0.76
	C	0.75	0.71	0.73	0.72	0.69	0.70	0.78	0.78	0.78
	D	0.78	0.73	0.75	0.73	0.65	0.69	0.79	0.74	0.76
Quantitative and Logical Ability	A	0.71	0.71	0.71	0.68	0.65	0.66	0.70	0.71	0.70
	B	0.76	0.69	0.72	0.70	0.62	0.66	0.72	0.71	0.71
	C	0.73	0.70	0.71	0.73	0.64	0.68	0.78	0.73	0.75
	D	0.77	0.73	0.75	0.71	0.68	0.69	0.76	0.74	0.75
Programming Skills	A	0.72	0.75	0.73	0.71	0.61	0.66	0.70	0.71	0.70
	B	0.79	0.71	0.75	0.69	0.64	0.66	0.71	0.70	0.70
	C	0.69	0.70	0.69	0.69	0.68	0.68	0.72	0.68	0.70
	D	0.76	0.74	0.75	0.72	0.67	0.69	0.76	0.75	0.75
Basic Skills	A	0.89	0.75	0.81	0.75	0.72	0.73	0.79	0.76	0.77
	B	0.79	0.72	0.75	0.71	0.69	0.70	0.79	0.76	0.77
	C	0.80	0.78	0.79	0.70	0.70	0.70	0.72	0.73	0.77
	D	0.81	0.79	0.80	0.78	0.75	0.76	0.81	0.80	0.80
Specialized Knowledge	A	0.89	0.86	0.87	0.87	0.81	0.80	0.88	0.83	0.85
	B	0.93	0.83	0.88	0.79	0.80	0.79	0.91	0.82	0.86
	C	0.90	0.89	0.89	0.81	0.82	0.81	0.83	0.81	0.82
	D	0.92	0.87	0.89	0.89	0.83	0.86	0.90	0.88	0.89
A = 50 distinct subjects, B = 50 distinct subjects + 10th+12th result										
C = 50 distinct subjects + subject classes, and D = 50 distinct subjects + 10th+12th result + subject classes										

Table 4.2: Ablation study (F-measure) for overall performance class assignment

Validation of Overall Performance: Additionally, we used the test dataset and processed it through three suggested modules—Naive Bayes, Logistic Regression, and their combination, to assess the system’s overall efficacy. In order to understand the significance of the suggested system, we have also carried out an ablation study on the test dataset’s features, as indicated in Table 4.2.

After that, in order to confirm the student’s overall performance classes—that is, their strengths and weaknesses—we compared the outcomes of the three ways. The results of the calculations for Precision, Recall, and F-measure are shown in Table 4.4.

In summary, our research indicates that classifying subjects facilitates the analysis of students’ expertise in specific domains. Furthermore, overall performance classes support the creation of educational apps like performance reports and career guides.

4.5 Observations

Building on previous research efforts, this chapter investigates the field of abstractive summarization. The first part of the chapter focuses on producing logical and instructive summaries through the use of deep learning techniques, including LSTM and RNN approaches. Works are done to explore the implementation of abstractive summarization using the

	M_{NB}			M_{LR}			$M_{Combined}$		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Soft Skills	0.78	0.73	0.75	0.73	0.65	0.69	0.79	0.74	0.76
Quantitative and Logical Ability	0.77	0.73	0.75	0.71	0.68	0.69	0.76	0.74	0.75
Programming Skills	0.76	0.74	0.75	0.72	0.67	0.69	0.76	0.75	0.75
Domain Knowledge	0.81	0.79	0.80	0.78	0.75	0.76	0.81	0.80	0.80
Advanced Knowledge	0.92	0.87	0.89	0.89	0.83	0.86	0.90	0.88	0.89

Table 4.3: Comparison of subject class assignment

Performance	Naïve Bayes			Logistic Regression			Combined		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
<i>Strength</i>	0.83	0.83	0.83	0.92	0.81	0.86	0.85	0.87	0.86
<i>Weakness</i>	0.75	0.75	0.75	0.83	0.76	0.79	0.79	0.80	0.79

Table 4.4: Evaluation metrics for overall student performance

encoder-decoder RNN architecture, a flexible sequence-to-sequence framework. In order to ensure the validity and applicability of the results obtained, these models go through rigorous training and testing using reliable open-source datasets like CNN/Daily Mails and DUC. These models emphasise on the complex nature of the sequence-to-sequence architecture by attempting to condense the semantic content of larger texts into shorter, coherent summaries. It is emphasised that careful consideration of input and output lengths is necessary to reduce padding and maintain important information.

The chapter additionally addresses difficulties with loss computation and output creation, as well as vocabulary size restrictions that were encountered during the construction of this deep learning-based model. In order to improve model performance and eliminate potential biases, strategies including sampling softmax and beam search techniques are described.

Furthermore, the chapter emphasises the significance of creating a student performance analysis system in the following sections, which moves to the application of natural language processing (NLP) techniques in educational technology. In-depth evaluations of pupils, especially in non-academic subjects and personal qualities, could be lacking in traditional examinations. Motivated by these difficulties, the research attempts to use NLP techniques to develop a performance analysis system that gives parents and teachers the ability to evaluate students' strengths and shortcomings in an efficient manner.

The development of a novel approach—using one-word abstractive summaries pro-

duced from a carefully selected performance analysis dataset—is a noteworthy contribution of this research. By reducing complicated numerical data on student performance into clear-cut and useful insights, this novel approach has the potential to provide educators and other stakeholders with essential knowledge for well-informed decision-making and intervention tactics.

Extensive tests are carried out on multiple datasets in order to assess the robustness and efficacy of the suggested solutions. The experimental framework includes metrics specifically designed to meet the goals of each methodology for dataset selection, model training, and assessment. The chapter attempts to show how effective the recommended methodologies are in producing abstractive summaries in various contexts by means of these thorough testing and analysis process.

In summary, the chapter highlights the need of developing abstractive summarization methods in the field of deep learning and utilising natural language processing (NLP) in educational technology to tackle intricate problems and enable significant understanding of student performance.

Chapter 5

Extractive to Abstractive Summarization

5.1 Introduction

In recent years, significant advancements have been achieved in the domain of natural language processing (NLP) with regard to automated summarization of texts. Summarization is essential for condensing extensive amounts of information into brief and informative summaries, which assist in understanding and making decisions in many fields. Historically, the discipline has been primarily influenced by two main methodologies; extractive summarisation, which involves the direct selection and presentation of significant excerpts from the source text; and, abstractive summarization, which entails the creation of summaries through the interpretation and rephrasing of the content. Researchers have been progressively concentrating on utilising extractive summarising outputs to improve the quality and fluency of abstractive summaries, notwithstanding the benefits and limits of each approach.

The reason for using extractive summarising outputs to generate abstractive summaries is due to the complimentary nature of these two approaches. Extractive techniques are highly effective in maintaining the original context and guaranteeing factual precision by directly choosing and integrating sentences from the source document. Nevertheless, their summaries frequently exhibit a deficiency in coherence and fluency, as they may have difficulties in successfully rephrasing or generalising material. In contrast, abstractive approaches provide the capability to produce summaries that resemble human language more closely. This is achieved by interpreting and rephrasing the information, resulting in summaries that are typically more succinct and easier to comprehend. Nevertheless, they

can encounter difficulties in choosing appropriate content and ensuring factual precision, especially when handling intricate or specialised texts.

Recently, academics have investigated different methods to connect extractive and abstractive summarization, with the goal of merging the advantages of both approaches while minimising their individual drawbacks. A noteworthy development in this domain involves the development of hybrid models, which integrate elements from extractive and abstractive approaches. These models frequently utilise pre-trained language representations, such as BERT (Devlin et al., 2018) or GPT (Radford et al., 2018), to extract semantic comprehension and context from the original material. In fact, (Liu et al., 2019) proposed a technique that employs pre-trained language models to enhance the quality of abstractive summaries in summarization challenges. This approach successfully utilises the semantic understanding embedded in the pre-trained representations.

Another emerging practice involves utilising reinforcement learning techniques to direct the production of abstractive summaries based on extractive outputs. In their study, Narayan et al. (2018a) introduced a methodology that utilises reinforcement learning to rate sentences during the extractive summarising phase. Then, abstractive summaries are generated in accordance with the selected sentences. By employing this approach, the model gains the capability to achieve an equilibrium between the amount of detail presented in extractive summaries and the elegance of abstractive summaries.

In addition, academics have investigated innovative structures and techniques for converting extractive results into abstractive summaries. Gu et al. (2016) suggested integrating a copying mechanism into sequence-to-sequence learning, enabling the model to directly replicate information from the original document while producing abstractive summaries. In a similar vein, Zhang et al. (2019) presented a saliency-driven abstractive summary technique that gives priority to significant information from the extractive output while producing coherent summaries.

In general, the current developments in utilising extractive summarization results to produce abstractive summaries emphasise the continuous endeavours to create more efficient and human-like summarization systems. This inspired us to work on the domain of producing abstractive summarising from extractive summarization as a starting point.

Utilising extractive summaries as a foundation for creating abstractive summaries offers numerous advantages. Initially, it can simplify the summarising work. The abstractive summarising system can concentrate on creating new sentences that convey the core message of the original content by utilising extractive summaries, instead of attempting to

include all the specifics from the original material. Secondly, it can enhance the quality of the produced summary. Extractive summaries condense the primary ideas and essential concepts from the original text, serving as a basis for creating new sentences. This can lead to a more cohesive and enlightening summary compared to creating one from the beginning. Utilising extractive summaries can enhance the efficiency of the summarization process. Extractive summarization requires less computational resources compared to abstractive summarization. Utilising extractive summaries as an initial step can decrease the processing needed to produce the ultimate summary in the abstractive summarization system.

More specifically, our primary focus is on developing systems that transition from extractive to abstractive approaches for summarizing tables found in scientific papers. In these papers, one of the most common tasks we face as researchers is the effective extraction of pertinent information from a variety of sources, such as tables, graphs, figures, and flowcharts. Scientific papers require tables in particular because they provide a clear and structured way to convey complicated data. The integration of presentation and content within tables, however, presents difficulties for conventional retrieval techniques. Consequently, it becomes essential to have concise summaries of table data. Researchers can save time and effort by using these summaries to quickly understand key material without having to read through complete publications. This motivated us to develop such systems and help the scientific community.

In this chapter, we outline our contributions to this field and propose three models for table data summarization. The initial model, which was T5 model based on transformers, was refined through the process of fine-tuning utilising our exclusive dataset in order to improve its performance. The second model proposed by us is a sequence-to-sequence (seq-to-seq) model. Finally, we propose the third model SETA, which is a similarity based encoder-decoder attentional based model. All of these models work on the philosophy of extractive to abstractive summary generation and produce summaries by selecting a pertinent extractive summary for each table and then generating an abstractive summary. Results suggest that our systems are capable of producing coherent summaries.

The subsequent sections of the chapter are organised as follows: The most recent developments in this area of research are covered in Section 5.2 The methods and models for extractive to abstractive summary creation are described in Section 5.3, 5.4, and 5.5. Finally the conclusion is presented in Section 5.6.

5.2 Survey

This section provides an overview of the current state of the literature concerning extractive to abstractive summarisation. As previously mentioned, extractive summarisation approaches seek to select pivotal sentences or phrases from the source material to provide the user with a summary. Proposed by [Mihalcea and Tarau \(2004\)](#), one of the pioneering algorithms in this field is TextRank, which uses PageRank like algorithms to identify important sentences according to their importance within the document graph. Another noteworthy method is LexRank, which was presented by [Erkan and Radev \(2004\)](#) and ranks sentences for extraction by calculating their cosine similarity.

In contrast, abstractive summarising approaches produce summaries by the interpretation and rewording of the original text, frequently with the use of natural language generation models. The [\(Sutskever and et al., 2014\)](#) sequence-to-sequence model, which uses Recurrent Neural Networks (RNNs) to map source sequences to target sequences, represents a significant achievement in this field. By more successfully capturing long-range dependencies, transformer-based architectures, like the one presented by [Vaswani et al. \(2017\)](#), have significantly enhanced abstractive summarization.

With the intention of enhancing the standard of produced summaries, recent research has concentrated on integrating the strengths of extractive and abstractive summarization systems. In a reinforcement learning approach that was proposed by [Gong et al. \(2018\)](#), the model extractively selects important sentences first, and then uses these selections to construct an abstractive summary. Similarly, [See et al. \(2017\)](#) proposed a Pointer-Generator network that successfully integrates extractive and abstractive techniques by dynamically determining whether to copy words from the original text or produce new words.

Furthermore, a crucial component of combining extractive and abstractive techniques has been the incorporation of attention mechanisms. A model that applies hierarchical attention to both words and sentences was presented by [Paulus et al. \(2017\)](#). By adopting this methodology, it is possible to generate abstractive summaries that preserve substantial details from the source material. In a similar vein, [Chen and Bansal \(2018\)](#) introduced a multi-level attention model that simultaneously pays to important words and phrases, enabling the creation of abstractive summaries with greater focus.

Moreover, recently the extraction and abstraction process has been optimised through the use of reinforcement learning as well. In contrast to conventional sequence-to-sequence models, [Narayan et al. \(2018b\)](#) presented a reinforcement learning-based technique that

learns to extract and abstract information from the source text simultaneously. Furthermore, a reinforcement learning strategy was presented by [Pasunuru and Bansal \(2018\)](#). In this technique, the model is rewarded for choosing informative words in the extractive phase and producing coherent summaries in the abstractive phase.

The use of hybrid designs to smoothly combine extractive and abstractive components has been investigated in other methods. A unified model that integrates extractive and abstractive modules into a single framework for end-to-end training and optimisation was presented by [Zhang et al. \(2019\)](#). Similar to this, [Liu and Lapata \(2019\)](#) presented a hierarchical neural network architecture that achieves state-of-the-art results on a variety of summarization tasks by simultaneously learning to choose relevant information and construct abstractive summaries.

Recently, models based on neural networks have been implemented in abstractive and extractive summarization techniques as well. [Conroy et al. \(2015\)](#), for example, presented a neural network model for extractive summarization that makes use of deep learning methods for phrase selection. Furthermore, [Nallapati et al. \(2017\)](#) achieved competitive performance on many datasets by using a convolutional neural network architecture for sentence extraction. These neural network-based techniques outperform conventional graph-based techniques in terms of accuracy and scalability.

Furthermore, graph-based methods have developed further, introducing increasingly complex sentence ranking algorithms. A graph-based approach for extractive summarization employing dynamic sentence weighting was presented by [Wan et al. \(2014\)](#). It adjusts to the unique features of the input document. Furthermore, [\(Zhang and Zhang, 2016\)](#) improved the representation of document semantics by the introduction of a graph-based method with sentence clustering, producing summaries that are more illuminating.

Research on extractive-abstractive summarization has also been accelerated by recent developments in pre-trained language models, such as BERT ([\(Devlin et al., 2019\)](#)) and GPT ([\(Radford et al., 2018\)](#)). These models can be refined for summary tasks, making use of their contextual language knowledge to provide more eloquent and informative summaries. These efforts have demonstrated potential for enhancing abstractive summaries' coherence.

It is important to remember that assessing hybrid extractive-abstractive summarization systems' efficacy might be difficult because a comprehensive evaluation is required. ROUGE is a frequently used statistic that calculates the overlap between summaries generated by the system and reference summaries based on n-gram recall. ROUGE offers

a numerical assessment of summary quality, although it could not adequately reflect the readability and semantic coherence of abstractive summaries.

In general, the combination of extractive and abstractive summarising methodologies has resulted in noteworthy progressions in automated summarization frameworks, affording more lucid and cohesive summaries throughout various fields. However, it should be emphasized that none of the works focus on table summarization from scientific papers. This motivated us to advance in this field.

5.3 Model 1 – Fine Tuned T5 Model

This section describes our very first approach of developing an extractive to abstractive summarization system of table based data.

5.3.1 Experimental Dataset

In Section 2.3, we presented our baseline dataset for developing table summarization systems, which includes 499 tables gathered from 200 computer science articles covering various areas such as Named Entity Recognition, Machine Translation, and Machine Learning. We expanded the dataset for our models by adding more publications collected from the ACL anthology corpus¹, resulting in a total corpus of 1500 papers, as discussed in Section 3.3.1.

Thereafter in Chapter 3, after preparing the dataset, we explained our method for creating extractive summaries for tables using a rule-based extractive summarising system, as outlined in Section 3.3.2. We suggested that a table usually has several extractive summaries, but generally just one abstractive summary. This hypothesis was also confirmed by validating our gold standard dataset, as depicted in Section 2.3.

Coming to our current approach, it includes using extractive summaries as a basis for creating abstractive summaries for tables in scientific papers. Thus, it was crucial to find the most appropriate extractive-abstractive summary pair for each table in the dataset while working on creating abstractive summaries from its extractive equivalents. This necessitated developing a system to provide the most appropriate extraction summary, a procedure detailed in Section 3.3.2.

Finally, the selected ideal extractive obtained as output from the rule-based system serves as a basis for our Fine-Tuned T5 Model, which is responsible for producing abstractive summaries. This enhances the efficiency and precision of the summarization process

¹<https://aclanthology.org/>

by enabling models to concentrate on finding crucial information in tables and choosing the most fitting extractive summary.

5.3.2 System framework

The T5 model, known as the "Text-to-Text Transfer Transformer," is a significant NLP architecture created by Google². This approach is designed to manage various text-related tasks, such as text summarization, and is based on the transformer framework. T5 differs from BERT-style models by consolidating all NLP tasks into a standardised structure, where input and output are represented as text strings. This text-to-text paradigm simplifies the use of a uniform model, loss function, and hyperparameters for various tasks such as summarization, machine translation, classification and question answering.

In our work, we utilized T5 small model for generating an abstractive summary. However we fine tuned this model and trained it using our table summarization dataset so as to get better results. The details of finetuning are discussed in the later sections. The inputs given to the model are the selected extractive summary for each table. The steps followed in utilizing this model are as follows:

Pre-processing the Text: In order to decide whether the input text could be used in the T5 model framework, we carefully reviewed it in this phase. This approach involves several crucial tasks to bring the textual data into accordance with the expectations and needs of the T5 model.

First, we worked on removing unnecessary details from the text input. This required locating and eliminating any extraneous or unrelated material that would obfuscate or take away from the text's primary point of emphasis.

Additionally, we concentrated on arranging the text in compliance with the unique demands of the T5 model. This included putting the textual data into a logical and structured manner so that it could be easily integrated and processed by the architecture of the model. Our goal in following the recommended formatting criteria was to guarantee uniformity with the input architecture of the T5 model.

Fine-tuning the Model: In order to fine tune the T5 model, we utilised a dataset with 4,800 data samples, which we will refer to as $(AB_i \rightarrow E_i)$, in which AB_i is the abstractive summary and E_i is the matching extractive summary that was chosen by a majority vote

²<https://blog.research.google/2020/02/exploring-transfer-learning-with-t5.html>

method that was described in previous sections. Our goal was to use this dataset to train the model so that it could produce abstractive summaries that faithfully reflected the main ideas of the extractive summaries that were supplied.

The extractive summary was the model’s input during the fine-tuning process, and the intended output was the matching abstractive summary. By employing this setup, the model underwent training to generate abstractive summaries that exhibited a high degree of similarity to the input extractive summaries in terms of both context and content.

To enhance the fine-tuning process, we employed a dropout rate as 0.1 and a continuous learning rate we used the score 0.001. These parameter settings made it easier for the model to find significant connections and patterns in the data by maintaining a continuous and effective training process.

In addition, we experimented with various input and output lengths to investigate how they affected the model’s functionality. Through this assessment, we were able to determine how altering the length parameters affected the model’s ability to produce accurate and logical abstractive summaries.

Summary Generation: After training, we evaluated the model’s performance using the remaining 1,210 selected extractive summary samples (AB_i). This assessment entailed inputting the samples into the model and examining the resulting output summaries to evaluate the performance of the developed model.

It’s crucial to remember that providing instructions to the T5 model for summarization necessitated a particular syntax for the input text. We included a specific job description, like “summarize:”, at the beginning of the input text to instruct the model to construct a summary. We demonstrated the formatting by using “summarize: [input text].”, with “[input text]” indicating the specific information to be summarised.

Introducing T5 to our exact this task description, guided the T5 model to concentrate on the exact NLP job to be done which is summarization. This allowed the model to comprehend and analyse the incoming data in a way that helps provide precise and pertinent summaries. We also made sure that the T5 model was correctly set up for summarization and could efficiently use the incoming data to provide logical and informative summaries. The next section describes the building of our next approach towards extractive to abstractive summarization of tables.

5.4 Model 2 – Seq-to-Seq Model

Our next approach is to create a sequence-to-sequence model with input consisting of a subset of extractive summaries from Section 3.3.2. The dataset utilised in this method is comparable to that which is mentioned in Section 5.3.1. The goal of this sequence-to-sequence (seq2seq) model is to produce abstractive summaries from extractive ones. It is based on sequence learning with neural networks. The model basically attempts to construct the goal sequence $Y = \{y_1, y_2, \dots, y_m\}$ as output after receiving a sequence $X = \{x_1, x_2, \dots, x_n\}$ as input.

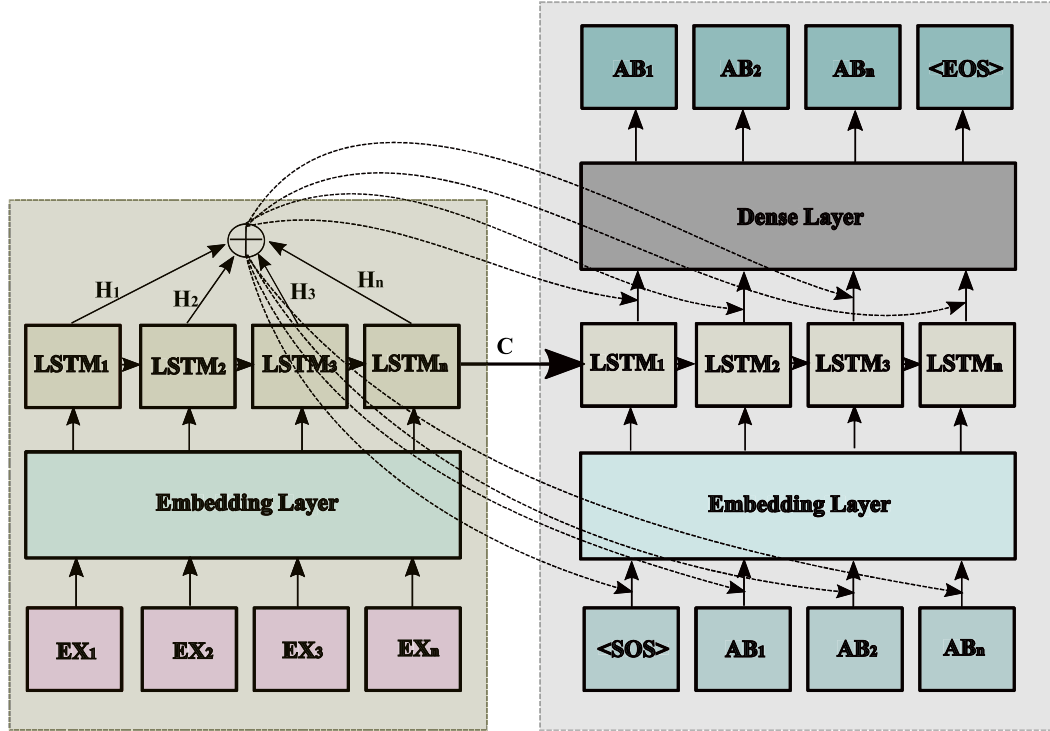


Figure 5.1: Architecture of the seq2seq model used to generate abstractive summaries from extractive summaries

where the input and target symbols, respectively, are x_i and y_i . The encoder and the decoder are the two components that make up the seq2seq model's architecture. We experimented with word-level embedding in a manner similar to the work of Mahata et al. (2019), and our model utilized the seq2seq architecture. Figure 5.1 illustrates how the seq2seq architecture operates at the word level. We used the Keras library for implementing the model.

Encoder: We utilized LSTM cells in the encoder design. One hot tensor of word-level embedded extractive summaries served as the cell’s input. The outputs from the encoder were deleted but the internal states of each cell were kept. This is done in order to maintain context-level information. The decoder cell was then given these states as beginning states.

Decoder: An LSTM cell was once more used for building the decoder, with initial states as the encoder’s hidden states. Sequences and states can both be returned by it. (Williams and Zipser, 1989) theory of ”teacher forcing” learning was applied in this instance. The decoder’s input consisted of a one hot tensor of abstractive summaries that were embedded at the word level. The target data was identical to the input but was offset one time step ahead of it. The decoder receives the information required for generation by using the initial states that were transmitted by the encoder. Consequently, given targets $[..., t]$ and an input sequence, the decoder can produce target data $[t + 1, ...]$. One word is anticipated at each output time step, which helps to forecast the output sequence.

Attention: The idea of attention Vaswani et al. (2017), inspired by human visual attention, removes the necessity of encoding the complete source sentence into a fixed-length vector by utilising the attention mechanism. The decoder prioritises various elements of the source text at different stages of output generation, enabling the model to determine which information to focus on depending on the input sequence and the current prediction status.

The context vector c_t is computed at each time step t by combining the source hidden states with weights.

$$c_t = \sum_{t=1}^{Tx} \alpha_t h_t$$

Each attention weight α_t indicates the relevance of the t -th source token x_t to the t -th target token y_t , computed as:

$$\alpha_t = \frac{1}{Z} \exp(\text{score}(E_y(y_{t-1}), s_{t-1}, h_t))$$

where

$$Z = \sum_{k=1}^{Tx} \exp(\text{score}(E_y(y_{t-1}), s_{t-1}, h_k))$$

The symbol Z denotes the normalisation constant. The $\text{score}()$ method uses a feed-forward neural network having a single hidden layer to evaluate the matching between the initial symbol T_x and the desired symbol y_t . s_t represents the target hidden state, and E_y represents the targeted embedding lookup table.

The parameters for training the model were as follows: the function used for activation was softmax, the optimizer was rmsprop, the number of epochs was 100, the batch size was 64, and the loss function was sparse categorical cross-entropy. A learning rate of 0.001 was used.

5.4.1 Experiments and Results

Since ROUGE and BLEU are the most widely used summarization metrics, we used them to evaluate the T5 model’s performance. Additionally, we have also evaluated the quality of the dataset by two other metrics namely adequacy and fluency [Hearne and Way \(2011\)](#), with the help of two linguists, familiar with the English language. The average values of the metrics are also reported. Adequacy measures how much of the created abstractive summary’s meaning is conveyed from the source summary. Fluency, on the other hand, shows how grammatically sound and easily understandable the generated summary phrase is to a native speaker. Fluency and adequacy are measured in the range of 1 through 5, with 1 denoting the lowest value and 5 the highest.

Table 5.1 displays the result. The pre-trained, fine-tuned T5 model outperformed the Seq-to-Seq model, as can be seen from the results. This is because the model could not be sufficiently trained due to the smaller amount of the training and testing datasets.

Metrics	Fine-tuned T5 Model	Seq-to-Seq Model
Avg_BLEU	58.2	16.25
Avg_ROUGE-1	0.36	0.21
Avg_ROUGE-L	0.31	0.18
Avg_Adequacy	4.02	2.07
Avg_Fluency	3.91	1.83

Table 5.1: Comparison between T5 and seq-to-seq model

5.5 Model 3 – SETA

In this section we present a novel method called SETA (Extractive to Abstractive summarising using a Similarity-Based Attentional Encoder-Decoder Model) after examining the two different text summarising models in the precious sections. SETA is a novel seq-to-seq similarity-based attentional encoder-decoder architecture which after training on the produced dataset, would learn to generate abstractive summaries from relevant extractive

summaries. This model aims to focus on important elements including saliency, adequacy, fluency, and coherence, which were neglected in prior table-based summarising systems.

The SETA architecture includes extractive summaries and their inter-sentential similarity embeddings, which are represented as an adjacency matrix of similarity scores. The scores are produced by different techniques, such as cosine similarity and Jaccard similarity. The SETA model use this method to capture the inherent connections between phrases in extractive summaries, making it easier to create abstractive summaries that are more logical and contextually relevant.

Rather than only extracting text without taking into account wider linguistic characteristics, this approach focuses on saliency to guarantee that the output summaries contain the most relevant information from the input extractive summaries. Adequacy is also a top priority to guarantee that the summaries cover the source information thoroughly. Additionally, the SETA model focuses on enhancing the fluency of generated summaries by utilising attention techniques that allow the model to focus on relevant parts of the input extracting summaries during the development process.

The upcoming sections describe the model development process in details.

5.5.1 Experimental Dataset

For the development of this model, we proposed an expanded version of the initial gold standard dataset used by the previous models 5.3 and 5.4. This extended model consisted of tables from over 2,600 papers we collected which spanned 20 different domains in computer science. The updated dataset statistics is shown in Table 5.3. Here, ESummary denotes extractive summary and ASummary denotes abstractive summary. $\#Eavg$ is the average number of extractive summaries present per table per paper in a particular paper type.

5.5.2 Dataset Quality Validation

Initially in the dataset, an abstractive summary AB_1 had multiple extractive summaries E_1, E_2 mappings denoted by $AB_i \rightarrow E_j$, where i is the total number of abstractive summaries and j is the total number of extractive summaries for each i . However, after selecting the most significant extractive summary for each table as discussed in the previous sections, we have made the dataset more relevant and compact. Next we have employed two methods for validating and evaluating the quality of the corpus namely, Inter Annotator agreement-based validation and Automatic Evaluation. The assessment process of

the corpus is briefly summarised in the ensuing subsections.

Inter Annotator agreement-based Validation To validate this dataset, we employed two human annotators, A_1 and A_2 , who were tasked with evaluating the mapping between an abstractive summary and the selected extractive summary for a particular table. Each annotator was tasked to identify whether the mappings were valid according to their opinion. A valid mapping was given a score of “1” and an invalid mapping was given a score of “0”. The dataset had 7815 tables and $AB_i \rightarrow E_j$ mappings which were validated by the annotators. Table 5.2 presents the confusion matrix constructed using the two annotators provided agreement-based scores for both labels (Valid – “1” and Invalid – “0”).

No. of Mappings ($AB_i \rightarrow E_j$) : 7815		Annotator 1	
		Valid (Score =1)	Invalid (Score=0)
Annotator 2	Valid (Score =1)	6953	110
	Invalid (Score=0)	100	652
Kappa Score		0.846	

Table 5.2: Inter annotator agreement analysis

These scores help to calculate the agreement between annotators A_1 and A_2 using Cohen’s Kappa³ agreement analysis approach.

Kappa coefficient score κ , which is defined in Equation (2) Viera et al. (2005), is used to illustrate the degree of agreement.

$$\kappa = \frac{Pr_a - Pr_e}{1 - Pr_e} \quad (5.1)$$

In this case, Pr_e denotes the proportion expected by chance, indicating a kind of random agreement between the annotators, and Pr_a represents the actual percentage that indicates full agreement between two annotators.

κ has a final value between -1 and 1, where 1 represents complete agreement, -1 represents entire disagreement, and 0 represents agreement by chance.

The analysis of agreement using Cohen’s Kappa, in this case, shows that for the abstractive to extractive mappings, the value of κ is 0.846 with an agreement of 96% confidence interval. A higher κ value indicates a stronger agreement.

³https://en.wikipedia.org/wiki/Cohen's_kappa

Paper Type	# Tables	Type: Text	Type: Numeric	ESummary		ASummary
				#Eavg	Elen_avg	Alen_avg
Automatic Summary	895	347	548	3	16	11
Machine Learning	845	423	422	4	15	12
Machine Translation	689	268	421	3	16	10
Named Entity Recognition	956	632	324	2	16	14
Question Answering	925	434	491	3	15	13
Sentiment Analysis	650	275	375	2	14	14
Speech Recognition	598	277	321	5	13	13
Text Classification	955	431	524	3	15	15
Text Segmentation	652	414	238	2	16	13
Word Sense Disambiguation	650	324	326	1	14	13
Total No. of papers	2600					

Table 5.3: Extended Dataset Statistics

Automatic Evaluation To further corroborate the findings of the external annotators, we then used two evaluation metrics: ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (Bilingual Evaluation Understudy). BLEU determines the level of similarity among the machine-generated summary and one or more reference summaries based on n-gram matching. On the other hand, the ROUGE family of assessment tools concentrates on recalling important information from the generated summary.

To do this, we determined the BLEU and ROUGE scores of the extractive summary in relation to its abstractive summary by selecting all 7,815 $AB_i \rightarrow E_j$ mappings. The most pertinent extractive summary, E_j , was selected as the candidate summary for this computation, while the AB_i served as the reference summary.

Table 5.4 reports the average BLEU and ROUGE-L (F1) scores for all combinations.

No. of Mappings : 7815			
Both Agree		A2 Agree	
Avg_BLEU	Avg_ROUGE-L	Avg_BLEU	Avg_ROUGE-L
93.1	0.65	51.5	0.45
A1 Agree		Both Disagree	
Avg_BLEU	Avg_ROUGE-L	Avg_BLEU	Avg_ROUGE-L
47.3	0.45	52.2	0.11

Table 5.4: An inter annotator agreement analysis to validate the dataset

After analyzing the results, we can come to the conclusion that the BLEU and ROUGE scores of the sample mappings that were agreed as VALID by both the annotators have higher values than the other combinations. This essentially supports our theory that the

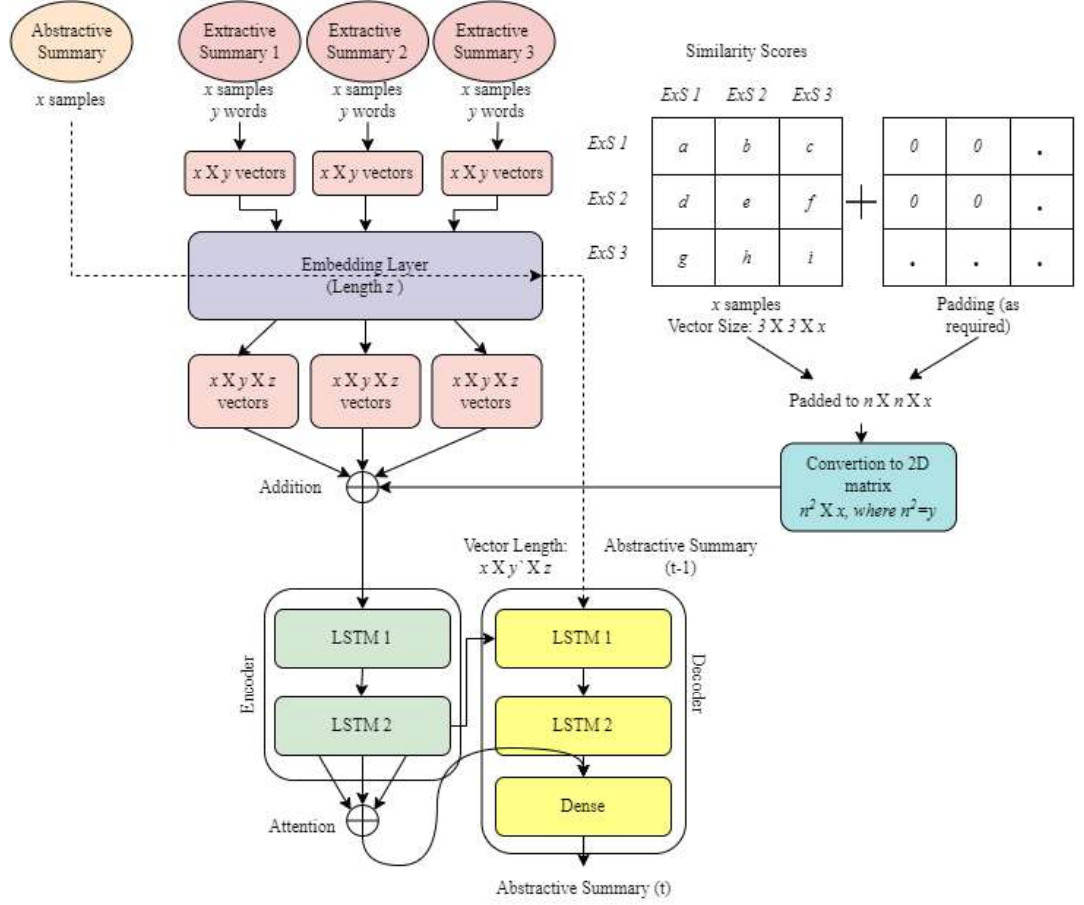


Figure 5.2: SETA – Architecture

summary samples serve as the best ones when both expert annotators are in agreement, demonstrating the datasets' quality.

After the dataset is validated we now move on to the details of the system development in the next sections.

5.5.3 Input Summary Selection

After the gold standard dataset is generated, we then used the system discussed in section 3.3.2 to select the most pertinent extractive summary for each table.

There is however a significant difference in the inputs provided to this model compared to the models described in 5.3 and 5.4. In the previous models we only selected the chosen extractive summary as input. In SETA however, our hypothesis is to take three sentences as input. They are the chosen extractive summary denoted as the reference sentence x , the sentence that comes before it ($x - 1$) and the sentence that comes after it ($x + 1$) in the paper text. This is to make sure we can take into consideration the contextual information

as much as possible in the output summary as the selected extractive summary (x) bears similarity to its preceding ($x - 1$) and subsequent sentences ($x + 1$).

To sum up, there are two components to this model: an encoder and a decoder. The encoder component of the model encodes a combination of input extractive summaries and inter-sentential similarity embeddings. This embedding is essentially an adjacency matrix consisting of similarity scores generated using multiple methods such as cosine and jaccard similarity.

After processing, the encoder creates a context vector with all of the sentence's information in it. The decoder portion is then initialised using this context vector, that acts as a language model that maps between abstractive summaries, differentiated by time frames (t and $t+1$) in our case.

5.5.4 Vectorisation of the summaries

In order to generate the vectors that will be given as input to our developed model, the relevant extractive summary that was selected in section by majority voting technique is considered. As explained this summary contains the reference sentence x , the sentence that comes before it ($x-1$) and the sentence that comes after it ($x+1$). This is done as we explored the hypothesis that the selected extractive summary (x) bears similarity to its preceding ($x-1$) and subsequent sentences ($x+1$). As a result, integrating these sentences as input is considered critical for generating context in the summary and achieving the saliency, non-redundancy, and fluency criteria within a coherent and organized framework. The extractive summary for every abstractive sample is transformed into an xXy vector, where y is the value denoting the average word count value of the extractive summary samples. The value of y was decided using a histogram plot that took into consideration the length of every extractive summary. The length at which the histogram plot had maximum weight was selected as the length of y . These extractive and abstractive text samples are then fed into a common embedding layer of length z . $xXyXz$ vectors for every extractive sample and abstractive sample represent the embedding layer output.

The process description is illustrated in Figure 5.2 and is given below.

5.5.5 Sentence embedding subspace

The aim of this model development is to enhance contextual information and highlight additional aspects in our abstractive summary. This is why we developed a sentence embedding subspace that was previously not explored in any previous works on summar-

ization. In order to do this, we computed a similarity vector of extractive summaries (3X3) for each abstract summary sample, as illustrated in Figure 5.2. Cosine similarity and Jaccard similarity are the similarity scores that we employed in our research. Thus, for x abstractive samples, the 3D vector 3X3X x is padded to $nXnXx$ which is then converted to a 2D vector $n^2 \times x$ where $y = n^2$. It can be observed that y , which is the average number of words in an extractive summary, has to be square of a natural number. Next, in order to concatenate the information to get better context, a matrix addition operation is performed between the 3D vector $xXyXz$ and the 2D vector n^2Xx . This operation ensures that the context from the entire sentence embedding space can be included into our resultant summary. This output then constitutes the input to the next part of our model. To generate abstractive summaries, we utilized a similarity matrix-based encoder-decoder model with attention. This model takes the chosen extractive summary as input and generates an abstractive summary and is described below.

5.5.6 Encoder-Decoder Model with Attention

Encoder: We utilized two layers of LSTM cells in the encoder design. The embedded extractive summary vector was obtained by concatenating the three extractive summary sentences along with the 3D similarity adjacency matrix created by calculating the similarity values between the three extractive summary sentences, which served as the cell’s input. This ensures the word-level, as well as sentence-level context features, are all included in the input to the encoder cells.

Encoder with Attention: In computational neuroscience, attention-related neural processes have been extensively researched. A lot of the inspiration for this idea comes from the way people focus their visual attention. In this approach, we use an attention method in place of encoding the complete source sentence into a fixed-length vector. This enables the decoder to produce output while focusing on various portions of the original text. Based on the input sequence and past predictions, the model learns which parts to prioritise.

Mathematically, the model calculates a context vector c_t at each time step (designated as ‘t’) as a weighted sum of the source’s hidden states::

$$c_t = \sum_{t=1}^{Tx} \alpha_t h_t \quad (5.2)$$

The significance of the t -th source token x_t to the t -th target token y_t is represented by each attention weight α_t , which is determined as follows:

$$\alpha_t = \frac{1}{Z} \exp(\text{score}(E_y(y_{t-1}), s_{t-1}, h_t)) \quad (5.3)$$

In this case, the normalisation constant is Z . Using a feed-forward neural network with a single hidden layer, the function $\text{score}()$ determines how similar the source symbol T_x and the target symbol y_t are to one another. s_t represents the target hidden state, and E_y represents the target embedding lookup table.

Decoder: The decoder was built using two LSTM cells that had been initialised with the encoder’s secret states. The decoder is capable of returning both sequences and states. (Williams and Zipser, 1989) established the concept of ”teacher forcing” learning, which was utilised in this case. The input to the decoder was a one-hot tensor of abstractive summaries embedded at the word level. Meanwhile, the target data mirrored the input, with a one-time step offset. The encoder passes on the initial states, which provide the information required for generation. As a result, the decoder can generate target data for time steps beyond t , indicated by $[t + 1, \dots]$, using the input sequence and previous target predictions up to time t . Each output time step predicts a single word, which results in the production of the complete output sequence.

The model was trained using the following parameters: rmsprop optimizer, softmax activation function, 64-batch size, 100 epochs, and sparse categorical cross-entropy loss. A learning rate of 0.001 was used.

Proposed Models	Avg. ROUGE-1	Avg. ROUGE-2	Avg. ROUGE-L	Avg. BLEU
Fine-tuned T5 Model	0.36	NA	0.31	58.2
Seq-to-Seq Model	0.21	NA	0.18	16.25
Embedding_WordVec + Similarity COSINE (SETA_v1)	0.34	0.42	0.29	38.51
Embedding_Glove + Similarity COSINE (SETA_v2)	0.31	0.36	0.27	36.7
Embedding_WordVec + Similarity Jaccard (SETA_v3)	0.32	0.34	0.21	32.21
Embedding_Glove + Similarity_Jaccard (SETA_v4)	0.26	0.36	0.13	33.23

Table 5.5: Automated Evaluation

5.5.7 Experiment and Results

Every experiment was conducted using our particular dataset, which is detailed in section 5.5.1. BLEU, ROUGE-1, ROUGE-2, ROUGE-L, and other traditional and standard measures were used to evaluate the effectiveness of the proposed approach. Table 5.5 displays the results of the automated evaluation. Furthermore, we have also conducted human evaluations on 50 random samples. Three participants were tasked with comparing

Models	Results		
	Average_Fluency	Average_Adequacy	Average_Saliency
Embedding_WordVec + Similarity COSINE (SETA_v1)	3.91	4.02	NA
Embedding_Glove + Similarity COSINE (SETA_v2)	1.83	2.07	NA
Embedding_WordVec + Similarity Jaccard (SETA_v3)	3.61	3.51	4.02
Embedding_Glove + Similarity_Jaccard (SETA_v4)	2.91	2.02	2.91
Embedding_WordVec + Similarity COSINE (SETA_v1)	3.20	3.02	2.95
Embedding_Glove + Similarity COSINE (SETA_v2)	2.82	3.12	2.54

Table 5.6: Human Evaluation Results

produced summaries against human-written summaries. They evaluated each summary using four criteria: (i) informativeness, (ii) salience, (iii) sentence coherence, and (iv) fluency and grammatical correctness.

These criteria are chosen for a specific purpose. Informativeness measures how much information the summary delivers, salience evaluates how well the summary fits with the original content, coherence assesses sentence flow, and fluency examines the summary’s grammatical quality. Every criterion received a score between 1 (worst) and 5 (best), depending on the category. Table 5.6 displays the average scores for each criterion. The outcomes have been compared with the two earlier models we proposed, which are discussed in sections 5.3 and 5.4, respectively. To ensure uniformity, it should be mentioned that each of these models were originally trained on our expanded dataset prior to the comparison.

Discussion As we can see in Tables 5.5 and 5.6, our model SETA_v1 designed with Word_to_Vec Embedding layer and Cosine similarity scores to prepare the similarity matrix, outperforms all the other models and is almost at par with the fine-tuned T5 model. All the other models also perform well. Thus we can definitely conclude that choosing relevant sentences which are in proximity of the relevant chosen summary definitely gives us better summaries. Though the margin looks small for some parameters like Rouge-1 and ROUGE-L it is quite substantial concerning the abstractive summary output. The primary reason behind this is because our developed dataset is insufficient for a deep learning model. However it can clearly be understood that the quality of the summary is not hampered due to the size of the dataset. Table 5, which presents the human evaluation findings, shows that the model SETA_v1, which uses a Word_to_Vec embedding layer and cosine similarity values, consistently outperforms both the current and past abstractive summary generation models.

5.6 Observations

Researchers have increasingly focused on using extractive summarisation outputs to improve the quality and fluency of abstractive summaries, acknowledging the complimentary nature of both approaches. Extractive approaches are excellent at retaining original context and assuring factual accuracy since they choose and integrate sentences directly from the source. They frequently lack clarity and fluency, and struggle to rephrase or generalise ideas. In contrast, abstractive algorithms provide summaries that are more closely related to human language, but they may struggle with content selection and factual correctness, particularly with complicated texts.

In this chapter, we investigated the transition from extractive to abstractive summarization, concentrating on tables presented in scientific papers. Tables are essential for communicating complex data, but they pose issues for traditional retrieval approaches. Our proposed models aim to build abstractive summaries by picking relevant extractive summaries for each table and then creating new summaries from them. In the course of our research, we initially proposed two distinct approaches for generating abstractive summaries from extractive ones: the fine-tuned T5 method and the sequence-to-sequence (seq2seq) method.

The first approach makes use of the T5 model, an important NLP architecture created by Google to handle a variety of text-related tasks, including text summarising. We fine-tuned the T5 model with our extensive dataset, tailoring it particularly to our summarization objective. This method involves feeding selected extractive summaries for each table into the T5 model, which subsequently produces abstractive summaries. This approach has the advantage of being resilient, as the T5 model has been pre-trained on a wide range of text-related tasks and can provide high-quality abstractive summaries.

In contrast, the sequence-to-sequence (seq2seq) technique uses neural networks to learn sequences. This method attempts to generate abstractive summaries by building the goal sequence from the input sequence of selected extractive summaries for each table. Unlike the T5 model, which is based on a pre-trained architecture, the seq2seq technique requires training a specific model for the summarising task. While this method may provide greater flexibility in model architecture and training parameters, it may also necessitate more prolonged training and tuning to achieve peak performance.

In our study, we examined the performance of these two methodologies using a variety of criterias, including ROUGE and BLEU for automated evaluation, as well as sufficiency and fluency, as assessed by English-speaking human evaluators. Our findings showed that

the fine-tuned T5 approach outperformed the seq2seq method across all criteria. This superiority can be attributed to the T5 model’s robustness and effectiveness, which was fine-tuned specifically for the summarising task.

However, it is worth mentioning that the seq2seq technique may have advantages in certain situations, such as when fine-grained control over model architecture and training parameters is required. Furthermore, more research and testing may be required to improve the performance of the seq2seq approach and fully realise its promise for abstractive summarization jobs.

Finally, we present the SETA model, which is a unique similarity-based attentional encoder-decoder architecture. This model stands out as it emphasises saliency, adequacy, fluency, and coherence, which were frequently disregarded in prior table-based summarising systems. SETA is a new sequence-to-sequence similarity-based attentional encoder-decoder architecture. It uses extractive summaries and their inter-sentential similarity embeddings, which are represented as an adjacency matrix of similarity scores. These ratings are calculated using a variety of methodologies, including cosine similarity and Jaccard similarity.

The SETA paradigm emphasises saliency, adequacy, fluency, and coherence, which were previously disregarded in table-based summarising systems. By capturing the natural links between terms in extractive summaries, SETA makes it easier to create contextually relevant and logically coherent abstractive summaries. Furthermore, it uses attention approaches to improve performance by allowing the model to concentrate on significant sections of the input extractive summaries during the summary production process.

The evaluation results confirm SETA’s performance, with SETA_v1 designed using Word_to_Vec Embedding layer and Cosine similarity scores to prepare the similarity matrix surpassing competing models in terms of both automated metrics and human review. Despite the restrictions of dataset size, our models consistently create high-quality summaries, demonstrating the possibility for using extractive summarization for abstract applications.

Chapter 6

Ranking

6.1 Introduction

Ranking problems are important in the field of Natural Language Processing (NLP) because they make it easier to efficiently organise and retrieve data from large textual databases. Ranking is the process of deriving an ordering over a list of things that maximises the list’s overall utility (Schutze et al., 2008). With applications ranging from web search and recommender systems to document summarization and question-answering, it is an important part of many information retrieval systems (Li, 2011).

Ranking initially emerged in research in the 1940s (Leontief, 1941), but it wasn’t until the end of the previous millennium (Page et al., 1999) that it became widely recognised as the basis for contemporary search engines.

NLP ranking has benefited greatly from a number of noteworthy studies conducted in the last few years. In one such work, (Yang et al., 2021) presented an effective ranking model that makes use of linguistic representations that have already been taught to increase the precision and effectiveness of information retrieval tasks. In question answering and document ranking challenges, Guo et al. (2020) presented a BERT-based ranking model that obtained state-of-the-art performance. In their investigation of deep learning techniques for NLP ranking, Liu et al. (2020) showed how well neural network architectures can capture the semantic links between queries and documents. Moreover, (Liu et al., 2021) presented a neural ranking model that extracts contextual data and long-term dependencies from text documents using self-attention mechanisms. (Zhang et al., 2021b) presented a cross-lingual ranking model that enhances cross-lingual retrieval of data by utilising multilingual embeddings. A transformer-based ranking model was introduced by (Kumar et al., 2021), which produced favourable outcomes in document

ranking challenges.

Additionally, in order to improve performance, [Chen et al. \(2021\)](#) studied retrieval-based ranking techniques, concentrating on the combination of retrieval models and neural ranking structures. [\(Zhou et al., 2020\)](#) investigated how contextual information affects ranking and showed how crucial contextual embeddings are for encapsulating text document semantics. Techniques for improving ranking performance by incorporating external knowledge sources were suggested by [\(Wang, 2020\)](#).

[\(Chen, 2020\)](#) investigated reinforcement learning methods for ranking in natural language processing, exhibiting the efficiency of reinforcement learning algorithms in refining ranking rules for retrieval of information assignments.

Together, these recent research reveal the wide variety of methods and strategies used in NLP ranking tasks, demonstrating the continuous efforts to increase the effectiveness and efficacy of information retrieval systems.

This chapter describes our participation in two important collaborative tasks that are mostly concerned with query ranking. We'll go over the models we created for these tasks in depth and provide the outcomes we got.

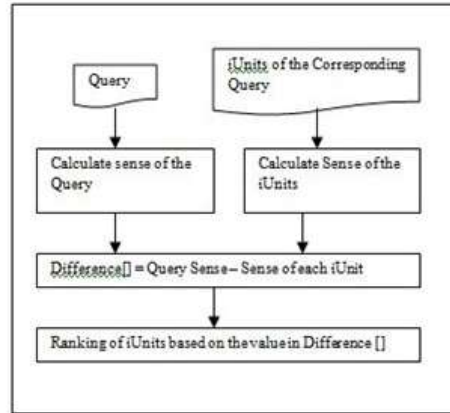
The rest of the chapter is organized as follows: [6.2](#) demonstrates our contribution to the NTCIR-12 Iunit ranking task; [6.3](#) shows our contribution to the IJCNLP shared task for ranking and finally [6.4](#) discusses and summarises the work done.

6.2 Model 1 – Sense Based Ranking

In this section, we delve into one of the key tasks within the domain of extractive summarization: the NTCIR-12 shared task. The NTCIR-12 shared task specifically focuses on two sub-tasks; ranking and summarization of web-based query outputs. In this chapter we mainly focus on our ranking model.

6.2.1 Experimental Dataset

For the ranking subtask, the NTCIR-12 organisers supplied both training and test data; for the summary subtask, they only had test data. Two files make up the test dataset for the ranking subtask: one file has 100 inquiries, and the second file has 4342 iUnits that are related to these queries. A detailed analysis of the dataset statistics is discussed in [Section 2.5.1](#) and the types of queries are shown in [Table 6.4](#)

**Figure 6.1:** Ranking Framework

6.2.2 IUnit Ranking

We developed a sense-based method to determine the order of information units. Determining the contextual similarity between the iUnits and the related inquiries is made easier by utilising the sentiment or information context. Specifically, we use sentiment lexicons to calculate the sense, which is the sense-based difference between the query and the iUnits' context. We next ranked these iUnits according to the differences in their senses. When the query "MC2-E-001" contains the term "hulk hogan," for example, the sense of the query is computed as 0.125. The difference between select iUnits and query sense is shown in Table 6.1. Table 6.2 shows our run file for the ranking subtask, which includes the query ID, matching iUnit ID, and their related ranks (sense-based). iUnits with a tighter sense match to the question are given a higher rank. The ranking structure used to ascertain the rank of iUnits for a given query is depicted in Figure 6.1. SentiWordNet and SenticNet sentiment lexicons help extract the tabulated evaluations' sense-based scores of iUnits, offering concept-based positive, negative, and neutral senses as well as the polarity scores associated with them.

QID	UID	DIFFERENCE
MC2-E-0002	MC2-E-0002-003	0.125
MC2-E-0002	MC2-E-0002-004	-0.125
MC2-E-0002	MC2-E-0002-005	0.0
MC2-E-0002	MC2-E-0002-005	-0.125

Table 6.1: Sense Difference Scores

QID	UID	RANK
MC2-E-0002	MC2-E-0001-003	1
MC2-E-0002	MC2-E-0001-004	18
MC2-E-0002	MC2-E-0001-005	12
MC2-E-0002	MC2-E-0001-006	18

Table 6.2: Run File for Ranking

6.2.3 Evaluation

For the ranking and summary portion of the NTCIR-12 Mobile-Click job (English), we have submitted two distinct run files. The normalised Discounted Cumulative Gain (nDCG) score for various cutoff criteria (k) is used to evaluate the iUnit ranking sub-task. How the evaluation metric Discounted Cumulative Gain (DCG) is calculated is shown here

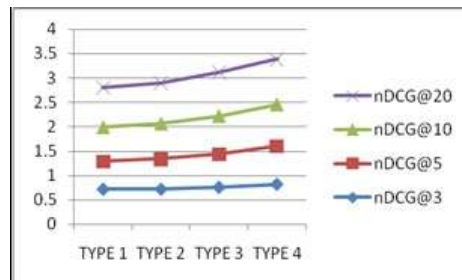
$$nDCG@K = \sum_{r=1}^K \frac{GG(u_r)}{\log 2(r+1)} \quad (6.1)$$

For calculating the iUnit ranting, the normalised form of DCG (nDCG) is

$$nDCG@K = \frac{DCG@K}{iDCG@K} \quad (6.2)$$

In order to validate the query's iUnit ranking result, another metric called Q-measure is implemented. Although nDCG is assessed using a graded technique based on ranked relevance, Q measure is generally recall based. The results of nDCG and Q-measure for the executions of the ranking part that we turned in are displayed in Table 6.3. For the purpose of evaluating our results depending on type, we have divided the given inquiries into four categories: individuals, places, random, and sentence type. Table 6.4 provides an example of these classes using the query id that was found in the test data.

The graphs in Figures 6.2 and 6.3, illustrate the mean nDCG and Q-measures values obtained for the diverse query kinds with various threshold levels.

**Figure 6.2:** Mean nDCG values for 4 Query Types at Varying Thresholds

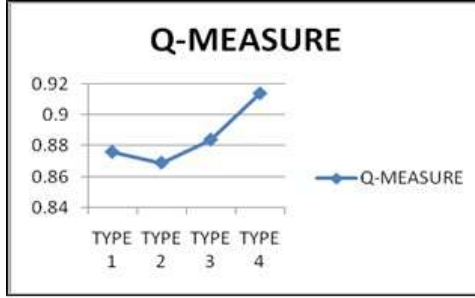


Figure 6.3: Mean Q measure values for 4 Query Types at Varying Thresholds

Query ID	Execution ID	nDCG@3	nDCG@5	nDCG@10	nDCG@20	Q
Mean	87	0.7012	0.7268	0.7807	0.8506	0.8859

Table 6.3: iUnit ranking system performance evaluation

6.3 Model 2 – Opinion Review Ranking

The goal of the IJCNLP-17 Review Opinion Diversification challenge is to rank the top k reviews of a product out of a large number of reviews. This facilitates the creation of a brief summary of every opinion expressed in those reviews. Subtasks A, B, and C make up the three sections of the task. The goal of each section is to choose the top- k reviews according to several standards, such as reviews’ representativeness, helpfulness, and exhaustiveness.

In order to summarise all of the opinions in the set, our tasks purpose is to select the top- k reviews for each product from a pool of reviews. These top- k reviews are chosen by the three subtasks using various techniques, taking into account representativeness, helpfulness, and exhaustiveness. Reviews that are evaluated as helpful are those that are deemed representative, which demonstrates shared viewpoints, and comprehensive, which indicates how well the reviews address thoughts about the product.

We developed three modules with a focus of representativeness, exhaustiveness, and usefulness in order to address subtasks A, B, and C. These modules aid in locating the top k reviews together with an estimate of their ranks. Naïve Bayes and Logistic Regression, two well-known classifiers, have been used in conjunction with features that were taken from the datasets. These features are customised based on the unique requirements of every subtask

6.3.1 Experimental Dataset

Section 2.5.3 discusses the experimental dataset used for this task in detail.

QID	TYPE
MC2-E-0002 - 020	Type 1 (People)
MC2-E-0022 - 040	Type 2 (Places)
MC2-E-0042 - 080	Type 3 (Random)
MC2-E-0083 - 010	Type 4 (Sentence)

Table 6.4: Query Classification

6.3.2 Subtask Description and Extraction of Feature

Subtask-A: The goal is to create a ranked list of k reviews based on their expected usefulness and to minimise repetition in the list.

The usefulness rating is a user-provided field in our data. We have discovered, via the analysis of the training and development datasets, that a review’s perceived usefulness is significantly influenced by specific language elements. In light of this, we extracted a number of features out of the datasets, such as word count, stop word count, bi-grams count, trigrams count, and tf-idf.

Subtask-B: In order to maximise the representation of distinct and various ideas represented in the reviews, the goal of this subtask is to build a ranked list of k reviews.

Following a thorough analysis of the dataset, we have shown that lexical characteristics of a review are critical in determining the optimal representation that encompasses widely held viewpoints. Thus, we have included three more features: the quantity of verbs, nouns, and adjectives from the datasets, in addition to the features utilised in Subtask-A. Adjectives aid in identifying sentiment keywords, while nouns and verbs aid in identifying significant language keywords.

Subtask-C: The goal of Subtask-C is to compile a ranked list that comprehensively addresses the majority of opinions regarding the product.

We have chosen to examine the feelings conveyed in each opinion in order to do this. We now know that presenting both favourable and unfavourable ideas broadens the coverage of different points of view. Furthermore, all types of opinions are captured by linguistic elements. In order to create the feature set, we added sentiment features, which include the quantity of sentiment words, negations, positive, and negative terms, in addition to the linguistic characteristics utilised in Subtask-B. Resources such as SenticNet ¹ and

¹<http://sentic.net/>

SentiWordNet ² have been used to extract sentiment features.

6.3.3 Modules Building

We used two popular supervised machine learning classifiers, Naïve Bayes and Logistic Regression, to predict the rank for each of the three subtasks. The features that were extracted and described in the preceding sections were used to train these classifiers. Next, we used the organisers’ test dataset to estimate the final rank for product reviews. We averaged the expected scores from both models to produce a single predicted rank for every review. The next steps describe the process:

Step-1: Multiple sentiment resources were used to process the development dataset that the organisers provided. This allowed us to extract a variety of features, including the quantity of nouns, verbs, negation words, and sentiment phrases for each subtask.

Step-2: Based on the characteristics of the subtasks, the retrieved attributes were categorised into three groups: representativeness, exhaustiveness, and helpfulness.

Step-3: These categories were then used to feed the Naïve Bayes and Logistic Regression classifiers, which helped to create each of the three modules in accordance.

Step-4: The rank of reviews for each module was then predicted using the test dataset that the organisers had provided.

Step-5: The top-k reviews (k chosen by the organisers as 5 and 10) for each product in the dataset were found with the use of the anticipated ranks.

The general procedure for evaluating each of the subtasks is explained in the section that follows.

Metrics	IJCINLP		BASE_R	
	5(List-size)	10(List-size)	5(List-size)	10(List-size)
cos	0.86	0.90	0.84	-
cos.d	0.87	0.91	0.84	-
cpr	0.71	0.68	0.74	-
a-dcg	4.98	5.71	4.53	-
wt	556.94	1384.6	533.41	-

Table 6.5: A comparison of all submitted modules of subtask-B

6.3.4 Evaluation

The results of our proposed modules were evaluated using metrics given to us by the task organisers as described in (Singh et al., 2017a). Standard metrics like cosine similarity

²<http://sentiwordnet.isti.cnr.it/s>

denoted as \cos and discounted cosine similarity denoted as \cos_d , alpha-DCG, cumulative proportionality denoted as cpr along with weighted relevance denoted as wt were used by us in subtask A. For B, we used nth (more than half) metric and subsequently for C, metrics like recall and unweighted relevance denoted as $unwt$ was used.

Subtask-A Validation: The appropriate nth metric is the proportion of reviews with over fifty percent of the votes given in favour of them. To be able to classify the favour as yes, no, or not counted, they have therefore computed *Upvotes*, or people who thought the review useful, and *Downvotes*, or people who did not. The computation of the nth can be exemplified by using equation 6.3, which considers the aggregate quantity of favours granted, as well as the total number of favours denied.

$$nth = \frac{yes}{yes + no} \quad (6.3)$$

where the values yes and no , stand for the total amount of favours that have been given, respectively.

For two distinct files with list sizes of 5 and 10, the equation helps determine the nth score of our suggested module Singh et al. (2017c). A comparative analysis of our module (JUNLP) and other modules used by participants in this shared work is presented in Table 6.6.

Groups	5(List-size)	10(List-size)
CYUT1	0.71	0.76
CYUT2	0.84	0.86
CYUT3	0.70	0.75
JUNLP	0.80	0.84
FAAD1	0.78	0.81
FAAD2	0.78	0.84
FAAD3	0.78	0.83

Table 6.6: Comparison of Subtask A participants

Validation of Subtask-B: Five different metrics have been used to evaluate Subtask-B. The metrics are alpha-DCG, discounted cosine similarity (\cos_d), cumulative proportionality (cpr), and weighted relevance. (wt)³. The scores of the metrics for our submitted modules are shown in Tables 6.7 and 6.5, which also offer a comparison with all other

³<https://sites.google.com/itbhu.ac.in/revopid-201fsu7/evaluation>

modules that were submitted for this subtask.

Metrics	5(List-size)	10(List-size)
cos	0.86	0.90
cos_d	0.87	0.91
cpr	0.71	0.68
a-dcg	4.98	5.71
wt	556.94	1384.6

Table 6.7: Subtask B assessment output

Metrics	List size 5	List size 10
unwt	10.94	28.93
recall	0.67	0.85

Table 6.8: Assessment results of module for subtask-C.

Validation of Subtask-C: Recall and unweighted relevance (unwt) are two additional metrics that are utilised to verify subtask-C’s output. Recall is the percentage of pertinent opinions that the ranking is able to successfully retrieve, while unweighted relevance represents a discounted total of the opinions in the ranked list. Table 6.8 presents the result of the proposed module for subtask-C.

Ultimately, we can say that, in comparison to other participants, our suggested modules offer scores for each of the three subtasks that are noticeable.

6.4 Observations

This Chapter reports our contribution to the ranking area of NLP. Our first contribution is our participation in NTCIR-12 task. We proposed a sense based ranking system for ranking iUnits with respect to queries. The IJCNLP-2017 RevOpiD task is addressed in this section with a rank prediction model for review opinion diversity. The work is divided into three smaller tasks: ranking of product reviews based on representativeness, exhaustiveness, and helpfulness. Three separate modules have been created for each of the subtasks. These modules have been designed by applying two popular machine learning classifiers, Naïve Bayes and Logistic Regression, on the retrieved features. With the assessment criteria that the organisers have supplied for each of the proposed modules, we are able to produce visible outcomes.

Chapter 7

Conclusion

We attempted to build resources and methods for both abstractive and extractive summarization in the current work. Our work on large-scale dataset preparations are presented in our thesis in **Chapter 2**. More precisely, we noticed that tables in scientific publications frequently include a variety of data, from category variables to numbers, thus it's critical to develop datasets that appropriately reflect this complexity. However, because careful annotation and validation are needed, creating such datasets is difficult. Inspired by this challenge, we set out to produce carefully structured and annotated datasets designed with table-based summary in consideration.

We presented two distinct datasets aimed at facilitating table-based summarization in this chapter. The first dataset focuses on condensing information from tables commonly found in academic research papers. This was a baseline dataset containing 499 tables from 200 scientific publications. This dataset went through a manual evaluation using agreement analysis which validated that our dataset quality was at par. Conversely, the second dataset evaluates student performance across different subject areas, pinpointing their strengths and weaknesses. These datasets were crafted to support the effective summarization of varied table content, catering to diverse contexts and needs. This dataset contained 2045 student samples which were then classified in 5 subject classes namely soft skills, logic and quantitative skills, basic skills, specialized knowledge and programming skills. This dataset also contained the students overall performance in these categories stating their strengths and weaknesses. With the help of two chosen professors, we also employed an extensive agreement analysis approach for validating the dataset for experimentation. A high agreement score denoted that the dataset quality was good. Furthermore, we actively participated in shared tasks such as IJCNLP 2017, NTCIR-12, and SCISUMM 2017 to advance our extractive summarization systems. This involve-

ment provided access to additional datasets and fostered collaborative research endeavors, which we discuss in this chapter. By leveraging diverse datasets like DUC, Opinosis, and CNN/Daily Mail, our aim is to enhance the efficacy, robustness, and applicability of our summarization models.

Chapter 3 showcases our significant contributions aimed at advancing the field of extractive summarization. We embarked on a thorough exploration of various extractive summarization techniques, focusing initially on summarizing tabular data from scholarly publications. Firstly we proposed an extended version of our gold standard table summarization dataset that we introduced in Chapter 2. This extended version consisted of tables from over 1500 research papers amounting to 6,010 tables, thereby increasing the volume of our gold standard dataset. We then introduced two extractive models for table content summarization. The first model, a rule based system, chose the most relevant extractive-abstractive summary pair with the help of standard assessment tools like ROUGE (ESSR), BLEU (ESSB) and LEXRANK (ESSL) and used a majority voting technique between them for selecting the most relevant extractive summaries. Next, we proposed two types of template-based models, TF-IDF and Transition point-based systems, that help to develop extractive summary templates of a table. Both of these models underwent strict validation steps through manual and automatic evaluation. Comparison of both the models, in cases where the summary pairs have been marked as valid summaries by both the external annotators, demonstrate that the rule-based approach is better than the template-based approaches.

Furthermore, we developed a sense-based model to rank and summarize English queries, as a participation in the NTCIR shared task. The data was provided as queries and iUnits, which are short summarized chunks of information about the queries. Intents were also provided which are some iUnits grouped together. A textrank algorithm along with wup similarity model was developed for summarization purpose. Next our involvement in the SCISUMM 2017 shared task provided an opportunity to apply extractive summarization techniques to scientific article summaries. There were three sub tasks, and we used standard similarity techniques like cosine similarity and tf-idf to develop three models. In our next task, we aimed to develop the summary of a document by selecting pertinent and significant sentences from the corpus. For this work, embedding techniques like word2vec and paragraph vectors were utilised to capture semantic similarities. The outcome summary quality was evaluated using the ROUGE assessment method. The dataset we utilised for our work is called 'Opinosis Dataset'. This dataset contains sentences col-

lected from user reviews about a specific topic. Additionally, rigorous methodologies for system evaluation, including automated metrics like BLEU and ROUGE, as well as inter-annotator agreement-based validation, provided critical insights into the effectiveness and performance of our methods. These evaluations guided our decision-making processes for future enhancements and advancements in extractive summarization techniques.

In **Chapter 4**, we have made contributions to the field of abstractive summarization. Firstly, we focused on employing deep learning techniques, specifically LSTM and RNN approaches, to develop models for abstractive summarization. These models were rigorously tested on well-established open-source datasets like DUC and CNN/Daily Mails, ensuring the reliability and precision of our findings. Our deep learning model, employing a sequence-to-sequence architecture, is designed to capture the semantic essence of longer texts and condense them into shorter, coherent sentences. During experimentation, we encountered challenges such as the potential bias towards unknown elements in cases where our vocabulary contained excessive unknown tokens. To mitigate this, we explored techniques like sampled softmax to speed up training and beam search to enhance output quality. Although our initial evaluation using ROUGE-L score yielded improvements, there is still room for enhancement as indicated by a score of 0.33. Additionally, we introduced an innovative approach that utilizes one-word abstractive summaries, derived from a carefully curated performance analysis dataset, to succinctly outline a student’s strengths and weaknesses. Furthermore, our evaluation included subject class identification using three classifiers, which demonstrated comparable performance. The ablation study table provided insights into the system’s performance with various features, all of which were rigorously validated through extensive measures.

In **Chapter 5**, our focus revolves around the development of table-based summarization systems, aimed at generating abstractive summaries of tables found in scientific articles using extractive summaries as a base.. We start by expanding the existing table summarization dataset to incorporate more training data, thereby enhancing the robustness of our models. We introduce three distinct models for the extractive-to-abstractive summarization task. The first model employs a transformer-based approach, specifically the fine-tuned T5 model using our customized expanded dataset. Evaluation of this model’s performance is conducted using well-established summarization metrics such as ROUGE and BLEU. Additionally, we assess the quality of our dataset using metrics like adequacy and fluency, with linguists familiar with the English language providing their insights. The findings indicate that the fine-tuned T5 model outperforms the sequence-

to-sequence model significantly across various metrics, illustrating its superiority in generating high-quality abstractive summaries. The discrepancy in performance is primarily attributed to the limited size of the dataset utilized for model training, highlighting the importance of continuously expanding the dataset to facilitate more effective model training and summary generation. The third model introduced, the SETA model, adopts an attentional-based encoder-decoder architecture, focusing on similarity-based approaches. This model leverages inter-sentential similarity embeddings derived from extractive summaries, represented as an adjacency matrix of similarity scores. By capturing inherent connections between phrases in extractive summaries, the SETA model facilitates the generation of more contextually relevant and logically coherent abstractive summaries. Results demonstrate that the SETA v1 model, incorporating Word to Vec Embedding layer and Cosine similarity scores for similarity matrix preparation, outperforms other models and exhibits performance comparable to the fine-tuned T5 model. Despite ongoing dataset development, the quality of summaries remains uncompromised, indicating promising prospects for further advancements in abstractive summarization.

Chapter 6 discusses our involvement in two significant collaborative endeavors primarily focused on query ranking. Firstly, we delve into our participation in the NTCIR-12 shared task, a prominent undertaking within the realm of extractive summarization. Specifically, we focus on the ranking and summarization of web-based query outputs, with our primary emphasis on the ranking subtask. Here, we detail our proposed model and the process of testing it using metrics provided by the organizers.

Following this, we explore our engagement in the IJCNLP-17 Review Opinion Diversification challenge. The objective of this challenge is to rank a product's top k reviews from a large pool of reviews, facilitating the creation of concise summaries of each opinion expressed. The challenge is subdivided into three tasks: A, B, and C. Each task aims to select the top-k evaluations based on various criteria, such as representativeness, helpfulness, and exhaustiveness. In this chapter, we document our efforts to address subtasks A, B, and C by developing three modules, each emphasizing a different aspect: representativeness, exhaustiveness, and utility.

Future research can expand the scope of our table-based summarization systems by incorporating multilingual datasets, enabling cross-lingual summarization capabilities. While our current models focus on English-language content, adapting them for non-English scientific tables would enhance their global applicability. Moreover, increasing the size and diversity of the training datasets—particularly for abstractive summarization—can signi-

ificantly improve model accuracy and generalization. Future efforts might also integrate large language models and reinforcement learning techniques to further refine summary quality and contextual accuracy. In addition, exploring user-centric evaluation methods, such as real-time feedback from domain experts, can offer valuable insights into system performance in practical applications. The SETA model and T5-based systems can also benefit from further architectural enhancements and hybridization with knowledge graphs to better capture semantic relationships within tables.

Bibliography

- Al-Radaideh, Q. A. and Bataineh, D. Q. (2018). A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms. *Cognitive Computation*, 10:651–669. [67](#)
- Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. (2017). Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*. [3](#), [4](#), [33](#), [67](#)
- An, C., Zhong, M., Chen, Y., Wang, D., Qiu, X., and Huang, X. (35:12498–12506, 2021). Enhancing Scientific Papers Summarization with Citation Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*. [13](#)
- Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations. To Appear*. [62](#)
- Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval*. Addison-Wesley. [3](#)
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*. [71](#)
- Bogdanova, D. and Loukachevitch, N. (2020). Text summarization in the age of transformer models: A survey. *arXiv preprint arXiv:2009.01349*. [4](#)
- Bougouin, A., Boudin, F., and Daille, B. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551. [35](#)
- Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117. [50](#)
- Brown, T. B. and et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*. [69](#)

- Burstein, J., Sabatini, J., and Shore, J. (2014). Natural language processing for educational applications. In *The Oxford Handbook of Computational Linguistics 2nd edition*. 22
- Carreras, X., Màrquez, L., and Padró, L. (2003). A simple named entity extractor using adaboost. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 152–155. 2
- Chen, a. n. h. (2020). Reinforcement learning for natural language processing. *Journal Name*, Volume Number(Issue Number):Page Numbers. 106
- Chen, D. L. and Mooney, R. J. (2008). Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*, pages 128–135. 15
- Chen, H.-H., Tsai, S.-C., and Tsai, J.-H. (2000). Mining tables from large scale html texts. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 166–172. Association for Computational Linguistics. 16
- Chen, L., Zhang, Z., Xie, R., and Zhou, G. (2021). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 2481–2491. 106
- Chen, Y.-C. and Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 675–686. 86
- Clark, K. and Manning, C. D. (2016). Deep reinforcement learning for mention-ranking coreference models. In *Empirical Methods on Natural Language Processing*. 2
- Conroy, J. M., Schlesinger, J. D., O’leary, D. P., Goldstein, J., and Goldstein, J. (2015). A neural network approach to context-sensitive generation of conversational summaries. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 52–60. 87
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 84
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019*

- Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. [3](#), [87](#)
- Devlin, J. and et al. (2023). Bert for summarization: Pre-training and fine-tuning strategies. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [2](#), [69](#)
- Dey, M., Mahata, S. K., and Das, D. (2023). Exploring summarization of scientific tables: Analysing data preparation and extractive to abstractive summary generation. *International Journal for Computers & Their Applications*, 30(4). [13](#)
- Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479. [35](#), [39](#), [86](#)
- Fabbri, A. R., Li, I., She, T., Li, S., and Radev, D. R. (2019). Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*. [4](#), [67](#)
- Florescu, C. and Caragea, C. (2017). Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1105–1115. [35](#)
- Gambhir, M. and Gupta, V. (2017). Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47:1–66. [33](#)
- Ganesan, K., Zhai, C., and Han, J. (2010). Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348. Association for Computational Linguistics. [14](#), [59](#)
- Gehrmann, S., Deng, K., and Rush, A. M. (2018). Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4098–4109. [70](#)
- Gliozzo, A., Magnini, B., and Strapparava, C. (2004). Unsupervised domain relevance estimation for word sense disambiguation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 380–387. [2](#)

- Gong, Y. and Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. [35](#)
- Gong, Y., Liu, X., and Zhang, Q. (2018). Reinforcement learning for extractive and abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2379. [86](#)
- Gu, Y., Li, J., Wang, S., and Liu, T. (2016). Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [84](#)
- Guo, J., Fan, Y., Ai, Q., Croft, W. B., and Feng, Y. (2020). A bert-based ranking model for question answering and passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 45–54. [105](#)
- Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States). [61](#)
- Hearne, M. and Way, A. (5(5):205–226, 2011). Statistical Machine Translation: A Guide For Linguists and Translators. *Language and Linguistics Compass*. [93](#)
- Hearst, M. A. (2015). Can natural language processing become natural language coaching? In *ACL (1)*, pages 1245–1252. [22](#)
- Hingu, D., Shah, D., and Udmale, S. S. (2015). Automatic text summarization of wikipedia articles. In *2015 international conference on communication, information & computing technology (ICCICT)*, pages 1–4. IEEE. [36](#)
- Jaidka, K., Chandrasekaran, M. K., Rustagi, S., and Kan, M.-Y. (2016). Overview of the 2nd computational linguistics scientific document summarization shared task (cl-scisumm 2016). In *Proceedings of the Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL 2016)*. [7](#), [14](#), [27](#)
- Jurafsky, D. and Martin, J. H. (2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson, 3rd edition. [3](#)

-
- Kaikhah, K. (2004). Automatic text summarization with neural networks. In *2004 2nd international IEEE conference on Intelligent Systems'. Proceedings (IEEE cat. No. 04EX791)*, volume 1, pages 40–44. IEEE. [36](#)
- Kato, M. P., Sakai, T., Yamamoto, T., Pavlu, V., Morita, H., and Fujita, S. (2016). Overview of the ntcir-12 mobileclick-2 task. In *NTCIR*. [7](#), [14](#), [25](#)
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632. [35](#)
- Kumar, M., Pandey, R., and Banerjee, S. (2021). Transformer-based ranking models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2114–2120. [105](#)
- Kyoomarsi, F., Khosravi, H., Eslami, E., Dehkordy, P. K., and Tajoddin, A. (2008). Optimizing text summarization based on fuzzy logic. In *Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008)*, pages 347–352. IEEE. [35](#)
- Lebret, R., Grangier, D., and Auli, M. (2016). Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*. [15](#)
- Leontief, W. W. (1941). The structure of american economy, 1919-1929. Technical report, Harvard University Press. [105](#)
- Lewis, M. and et al. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*. [69](#)
- Li, H. (2011). Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 4(1):1–113. [105](#)
- Li, Z., Liu, X., Chen, Y., and Sun, M. (2021). DRGAT: A Dual-Reading Graph Attention Network for Abstractive Summarization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 142–156. [70](#)
- Liang, P., Jordan, M. I., and Klein, D. (2009). Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 91–99. [15](#)

- Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics. [63](#)
- Liu, M. et al. (2022). Tablebert: Pre-training for table understanding and generation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-22)*. [13](#)
- Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*. [33](#)
- Liu, S., Cao, J., Yang, R., and Wen, Z. (2023). Long text and multi-table summarization: Dataset and method. *arXiv preprint arXiv:2302.03815*. [15](#)
- Liu, X., He, P., Chen, W., Gao, J., and Liu, H. (2021). Neural ranking models with weak supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1844–1854. [105](#)
- Liu, Y. and Lapata, M. (2019). Hierarchical transformers for multi-document summarization. *arXiv preprint arXiv:1905.13164*. [33](#), [87](#)
- Liu, Y. and Lapata, M. (2020). PreSumm: A Neural Model for Abstractive Text Summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3142–3152. [69](#)
- Liu, Y., Lapata, M., and Li, F. (2019). Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3730–3740. [70](#), [84](#)
- Liu, Y., Ren, Z., Shen, Y., and Zhao, J. (2020). Deep reinforcement learning for click-through rate prediction with attention-based candidate sampling. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 355–364. [105](#)
- López, F. R., Jiménez-Salazar, H., and Pinto, D. (2007). A Competitive Term Selection Method for Information Retrieval. In *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, pages 468–475. [41](#)
- Ma, Q. and Sakti, S. (2020). Incorporating External Knowledge into Pre-trained Transformers for Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1–13. [70](#)

- Mahata, S. K., Garain, A., Rayala, A., Das, D., and Bandyopadhyay, S. (2019). A Hybrid Approach To Machine Translation For Lithuanian To English. *2019 Fourth Conference on Machine Translation*, pages 283–286. [91](#)
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. [3](#)
- Mashechkin, I. V., Petrovskiy, M., Popov, D., and Tsarev, D. V. (2011). Automatic text summarization using latent semantic analysis. *Programming and Computer Software*, 37:299–305. [35](#)
- McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. [29](#)
- Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411. [50](#), [86](#)
- Nallapati, R., Zhai, F., and Zhou, B. (2016). Abstractive Text Summarization using Sequence-to-Sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 280–290. [70](#)
- Nallapati, R., Zhou, B., Gulcehre, C., Xiang, B., Chandar, S., and Bengio, Y. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. [87](#)
- Narayan, R., Cohen, S. B., and Lapata, M. (2018a). Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. [84](#)
- Narayan, S., Cohen, S. B., and Lapata, M. (2018b). Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807. [86](#)
- Nenkova, A. and McKeown, K. (2011). Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2-3):103–233. [2](#), [4](#)

- Ng, H. T., Lim, C. Y., and Koo, J. L. T. (1999). Learning to recognize tables in free text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 443–450. Association for Computational Linguistics. [16](#)
- Nogaito, I., Yasuda, K., and Kimura, H. (2016). Study on automatic scoring of descriptive type tests using text similarity calculations. In *EDM*, pages 616–617. [22](#)
- Ozsoy, M. G., Alpaslan, F. N., and Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science*, 37(4):405–417. [35](#)
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bring order to the web. Technical report, Technical report, stanford University. [61](#)
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab. [105](#)
- Pasunuru, R. and Bansal, M. (2017). Reinforced Video Captioning with Entailment Rewards. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 979–985. [69](#)
- Pasunuru, R. and Bansal, M. (2018). Multi-task learning for extractive and abstractive summarization using sentence-level supervision. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4481–4492. [87](#)
- Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*. [86](#)
- Paulus, R., Xiong, C., and Socher, R. (2018). A Deep Reinforced Model for Abstractive Summarization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, pages 1–13. [70](#)
- Radev, D., Hovy, E., and McKeown, K. (2002). Introduction to the special issue on summarization. *Computational linguistics*, 28(4):399–408. [67](#)
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. Technical report, OpenAI Technical Report. [84](#), [87](#)
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9. [2](#)

- Raffel, C. and et al. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*. [69](#)
- Raphal, N., Duwarah, H., and Daniel, P. (2018). Survey on abstractive text summarization. In *2018 international conference on communication and signal processing (ICCSP)*, pages 0513–0517. IEEE. [68](#)
- Reiter, E. and Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87. [2](#)
- Reynolds, D. A. and Rose, R. C. (1995). Robust text-independent speaker identification using gaussian mixture speaker models. *IEEE transactions on speech and audio processing*, 3(1):72–83. [2](#)
- Rieger, B. B. (1991). *On distributed representation in word semantics*. International Computer Science Institute Berkeley, CA. [34](#)
- Sankarasubramaniam, Y., Ramanathan, K., and Ghosh, S. (2014). Text summarization using wikipedia. *Information Processing & Management*, 50(3):443–461. [35](#)
- Schutze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to Information Retrieval*, volume 39. Cambridge University Press, Cambridge. [105](#)
- Schwind, C. (1988). Sensitive parsing: error analysis and explanation in an intelligent language tutoring system. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 608–613. Association for Computational Linguistics. [22](#)
- See, A., Liu, P. J., and Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1073–1083. [69](#), [70](#), [86](#)
- Singh, A. K., Thawani, A., Gupta, A., and Mundotiya, R. K. (2017a). Evaluating opinion summarization in ranking. In *Proceeding of the 13th Asia Information Retrieval Societies Conference (AIRS 2017)*, Jeju island, Korea. [111](#)
- Singh, A. K., Thawani, A., Panchal, M., Gupta, A., and McAuley, J. (2017b). Ijcnlp-2017 task 3: Review opinion diversification (revopid-2017). *Proceedings of the IJCNLP 2017, Shared Tasks*, pages 17–25. [7](#), [14](#), [29](#)
- Singh, A. K., Thawani, A., Panchal, M., Gupta, A., and McAuley, J. (2017c). Overview of the ijcnlp-2017 shared task on review opinion diversification (revopid-2017). In *Proceedings of the IJCNLP-2017 Shared Tasks*, Taipei, Taiwan. AFNLP. [112](#)

- Steinberger, J., Jezek, K., et al. (2004). Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4(93-100):8. [36](#)
- Suanmali, L., Binwahlan, M. S., and Salim, N. (2009a). Sentence features fusion for text summarization using fuzzy logic. In *2009 Ninth International Conference on Hybrid Intelligent Systems*, volume 1, pages 142–146. IEEE. [35](#)
- Suanmali, L., Salim, N., and Binwahlan, M. S. (2009b). Fuzzy logic based method for improving text summarization. *arXiv preprint arXiv:0906.4690*. [35](#)
- Suleiman, D. and Awajan, A. (2020). Deep learning based abstractive text summarization: approaches, datasets, evaluation measures, and challenges. *Mathematical problems in engineering*, 2020:1–29. [68](#)
- Sunitha, C., Jaya, A., and Ganesh, A. (2016). A study on abstractive summarization techniques in indian languages. *Procedia Computer Science*, 87:25–31. [4](#), [67](#), [68](#)
- Sutskever, I. and et al. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NeurIPS 2014)*. [70](#), [86](#)
- Svore, K., Vanderwende, L., and Burges, C. (2007). Enhancing single-document summarization by combining ranknet and third-party sources. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 448–457. [36](#)
- Tas, O. and Kiyani, F. (2007). A survey automatic text summarization. *PressAcademia Procedia*, 5(1):205–213. [3](#), [33](#)
- Tengli, A., Yang, Y., and Ma, N. L. (2004). Learning table extraction from examples. In *Proceedings of the 20th international conference on Computational Linguistics*, page 987. Association for Computational Linguistics. [16](#)
- Urbizagástegui, R. (1999). Las posibilidades de la ley de zipf en la indización automática. *Reporte de la Universidad de California Riverside*. [41](#)
- Van Halteren, H., Zavrel, J., and Daelemans, W. (2000). Improving accuracy in nlp through combination of machine learning systems. *Computational Linguistics*, 5:2071–2074. [2](#)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is All You Need. In *Advances in Neural Information Processing Systems*, pages 6000–6010. [70](#), [86](#), [92](#)

- Viera, A. J., Garrett, J. M., et al. (2005). Understanding interobserver agreement: the kappa statistic. *Fam Med*, 37(5):360–363. [24](#), [42](#), [95](#)
- Wan, X., Xiao, J., Guo, J., Xiao, J., Lu, H., and Yu, N. (2014). Graph-based multimodality fusion for extractive news summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247. [87](#)
- Wang, a. n. h. (2020). Enhancing natural language processing tasks with graph neural networks. *Journal Name*, Volume Number(Issue Number):Page Numbers. [106](#)
- Wang, M., Wang, X., and Xu, C. (2005). An approach to concept-obtained text summarization. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.*, volume 2, pages 1337–1340. IEEE. [35](#)
- Wang, Y. and Hu, J. (2002). A machine learning based approach for table detection on the web. In *Proceedings of the 11th international conference on World Wide Web*, pages 242–250. ACM. [16](#)
- Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280. [92](#), [100](#)
- Wiseman, S., Shieber, S. M., and Rush, A. M. (2017). Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*. [16](#)
- Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. *arXiv preprint cmp-lg/9406033*. [50](#), [52](#)
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2021). Efficient attention: Attention with linear complexities. In *International Conference on Learning Representations (ICLR)*. [105](#)
- Yeh, J.-Y., Ke, H.-R., Yang, W.-P., and Meng, I.-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information processing & management*, 41(1):75–95. [36](#)
- Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75. [2](#)

- Zhang, J. et al. (2021a). Tabularbert: Pre-training for tabular data understanding and generation. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2021)*. [13](#)
- Zhang, M. and Zhang, Y. (2016). Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 484–494. [87](#)
- Zhang, X., Lapata, M., Wei, F., and Zhou, M. (2018). Neural latent extractive document summarization. *arXiv preprint arXiv:1808.07187*. [68](#)
- Zhang, X., Wang, M., Gong, Y., and Huang, X. (2019). Saliency-driven abstractive summarization on document streams. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. [84](#), [87](#)
- Zhang, Y., Ott, M., Hu, W., Chen, H., Huang, Y., Hsieh, C. J., and Chen, L. (2021b). Cross-lingual information retrieval with multilingual knowledge. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1855–1866. [105](#)
- Zhou, Q., Yang, N., Wei, F., Tan, C., Bao, H., and Zhou, M. (2018). Neural Document Summarization by Jointly Learning to Score and Select Sentences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 654–663. [70](#)
- Zhou, Y., Dai, Z., and Dong, L. (2020). Contextualized graph-based reasoning for document-level relation extraction. In *Proceedings of the Conference*. [106](#)