

Text Summarization: Models, Methods and Evaluation

Thesis Submitted by

Tohida Rehman

Doctor of Philosophy (Engineering)

Department of Information Technology
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India
2024

Text Summarization: Models, Methods and Evaluation

by

Tohida Rehman

Registration Number: 10222090001

Thesis submitted for the

Doctor of Philosophy (Engineering)

Degree of Jadavpur University, Kolkata, India

Supervisors:

Prof. Samiran Chattopadhyay

Professor(R)

Dept. of Information Technology

Jadavpur University, Salt Lake Campus

Kolkata-700106

West Bengal

India

Dr. Debarshi Kumar Sanyal

Assistant Professor

School of Mathematical & Computational Sciences

Indian Association for the Cultivation of Science

Kolkata-700032

West Bengal

India

2024

Jadavpur University

Kolkata 700 032, India

INDEX NO. 54/22/E

Title of the thesis :

Text Summarization: Models, Methods and Evaluation

Name, Designation and Institution of the Supervisors:

Prof. Samiran Chattopadhyay

Professor(R)

Department of Information Technology

Jadavpur University, Salt Lake Campus

Kolkata-700106

West Bengal

India

Dr. Debarshi Kumar Sanyal

Assistant Professor

School of Mathematical & Computational Sciences

Indian Association for the Cultivation of Science

Kolkata-700032

West Bengal

India

List of Publications

Journal papers

1. **Tohida Rehman**, Debarshi Kumar Sanyal, Samiran Chattopadhyay, Plaban Kumar Bhowmick, and Partha Pratim Das. Generation of Highlights From Research Papers Using Pointer-Generator Networks and SciBERT Embeddings. *IEEE Access*, Volume 11, pages 91358–91374, 2023, DOI:<https://doi.org/10.1109/ACCESS.2023.3292300>.
2. **Tohida Rehman**, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. Research Highlight Generation with ELMo Contextual Embeddings. *Scalable Computing: Practice and Experience*, Volume 24, No. 2, pages 181–190, 2023, DOI: <https://doi.org/10.12694/scpe.v24i2.2238>.

Presentations and Proceedings in International/National Conferences/Workshops

1. **Tohida Rehman**, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. Can pre-trained language models generate titles for research papers? Accepted for presentation at the International Conference on Asia-Pacific Digital Libraries (ICADL 2024).
2. **Tohida Rehman**, Ronit Mandal, Abhishek Agarwal, and Debarshi Kumar Sanyal. Hallucination Reduction in Long Input Text Summarization. In *Proceedings of International Conference on Security, Surveillance and Artificial Intelligence: ICSSAI-2023*, page 307-316. CRC Press, 2024. eBook ISBN 9781003428459.
3. **Tohida Rehman**, Samiran Chattopadhyay, and Debarshi Kumar Sanyal. Abstractive summarization of scientific documents: Models and Evaluation Techniques. In *Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE'2023*, page 121–124, New York, NY, USA, 2024. Association for Computing Machinery. <https://doi.org/10.1145/3632754.3632771>.
4. **Tohida Rehman**, Debarshi Kumar Sanyal, Prasenjit Majumder, and Samiran Chattopadhyay. Named Entity Recognition Based Automatic Generation of Research Highlights. In *Proceedings of the Third Workshop on Scholarly Document Processing (SDP 2022) collocated with COLING 2022*, pages 163–169, Gyeongju, Republic of Korea, October 2022. Association for Computational Linguistics.
5. **Tohida Rehman**, Suchandan Das, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. An Analysis of Abstractive Text Summarization Using Pre-trained

Models. In Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2021, pages 253–264. Springer, 2022.

6. **Tohida Rehman**, Debarshi Kumar Sanyal, Samiran Chattopadhyay, Plaban Kumar Bhowmick, and Partha Pratim Das. Automatic Generation of Research Highlights from Scientific Abstracts. In Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) collocated with JCDL 2021, pages 69–70, 2021.

Under review in International Conferences/Journal

1. **Tohida Rehman**, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. SilverCSPicoSum: A Dataset of Very Short Summaries Generated with ChatGPT-3.5.

List of Patents: Nil

List of Presentations in National/International Conferences/Workshops:

1. **Tohida Rehman**, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. Can pre-trained language models generate titles for research papers? Accepted for presentation at the International Conference on Asia-Pacific Digital Libraries (ICADL 2024).
2. **Tohida Rehman**, Ronit Mandal, Abhishek Agarwal, and Debarshi Kumar Sanyal. Hallucination Reduction in Long Input Text Summarization. In Proceedings of International Conference on Security, Surveillance and Artificial Intelligence: ICSSAI-2023, page 307-316. CRC Press, 2024. eBook ISBN 9781003428459.
3. **Tohida Rehman**, Samiran Chattopadhyay, and Debarshi Kumar Sanyal. Abstractive summarization of scientific documents: Models and Evaluation Techniques. In Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, FIRE'2023, page 121–124, New York, NY, USA, 2024. Association for Computing Machinery. <https://doi.org/10.1145/3632754.3632771>.
4. **Tohida Rehman**, Debarshi Kumar Sanyal, Prasenjit Majumder, and Samiran Chattopadhyay. Named Entity Recognition Based Automatic Generation of Research Highlights. In Proceedings of the Third Workshop on Scholarly Document Processing (SDP 2022) collocated with COLING 2022, pages 163–169, Gyeongju, Republic of Korea, October 2022. Association for Computational Linguistics.
5. **Tohida Rehman**, Suchandan Das, Debarshi Kumar Sanyal, and Samiran Chattopadhyay. An Analysis of Abstractive Text Summarization Using Pre-trained Models. In Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2021, pages 253–264. Springer, 2022.
6. **Tohida Rehman**, Debarshi Kumar Sanyal, Samiran Chattopadhyay, Plaban Kumar Bhowmick, and Partha Pratim Das. Automatic Generation of Research Highlights from Scientific Abstracts. In Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021) collocated with JCDL 2021, pages 69–70, 2021.

PROFORMA – 1
“Statement of Originality”

I, **Tohida Rehman**, registered on **26/04/2022** do hereby declare that this thesis entitled **“Text Summarization: Models, Methods and Evaluation”** contains literature survey and original research work done by the undersigned candidate as part of Doctoral studies.

All information in this thesis have 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 **2%**.

Signature of Candidate:

Tohida Rehman-----

(Tohida Rehman)

Date : 25/09/2024

Certified by Supervisors:

(Signature with date, seal)

1. Samiran Chattopadhyay 25/09/2024
(Samiran Chattopadhyay)

PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block -LB, Plot-8, Sector-3
Salt Lake, Kolkata-700106, India

2. Debarshi Kumar Sanyal 25/09/2024
(Debarshi Kumar Sanyal)

DR. DEBARSHI KUMAR SANYAL
Assistant Professor
School of Mathematical and Computational Sciences
Indian Association for the Cultivation of Science
Kolkata-700 032, India

PROFORMA – 2

“CERTIFICATE FROM THE SUPERVISORS”

This is to certify that the thesis entitled “**Text Summarization: Models, Methods and Evaluation**” submitted by **Ms. Tohida Rehman**, who got her name registered on **26/04/2022** for the award of Ph.D. (Engg.) degree of Jadavpur University is absolutely based upon her own work under the supervision of **Prof. Samiran Chattopadhyay**, Department of Information Technology, Jadavpur University, Kolkata and **Dr. Debarshi Kumar Sanyal**, School of Mathematical & Computational Sciences, Indian Association for the Cultivation of Science and that neither her thesis nor any part of the thesis has been submitted for any degree/diploma or any other academic award anywhere before.

Signatures of the Supervisors with Date and Official Seal

Samiran Chattopadhyay 25/09/2024

Prof. Samiran Chattopadhyay

Professor(R),
Department of Information Technology
Jadavpur University,
Block– LB, Plot– 8, Sector–3,
Salt Lake, Kolkata 700106, India

PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block -LB, Plot-8, Sector-3
Salt Lake, Kolkata-700106, India

Debarshi Kumar Sanyal 25/09/2024

Dr. Debarshi Kumar Sanyal

Assistant Professor,
School of Mathematical & Computational Sciences,
Indian Association for the Cultivation of Science,
Kolkata-700032, India

DR. DEBARSHI KUMAR SANYAL
Assistant Professor
School of Mathematical and Computational Sciences
Indian Association for the Cultivation of Science
Kolkata-700 032, India

Acknowledgements

My doctoral journey has offered me an enriching experience that involves persistence, commitment, and determination. This journey has been a true adventure, teaching me to face the challenges of the “troughs” with courage and to remain humble during the “crests”. While the path has been challenging, the support of a few remarkable individuals has made it a wonderfully memorable experience. I sincerely acknowledge the contributions of everyone who has been part of this journey.

Firstly, I would like to express my gratitude to my supervisors, Prof. Samiran Chattopadhyay and Dr. Debarshi Kumar Sanyal. Not only did they guide me throughout this journey, but they also believed in me at every step. They helped me choose my research domain, encouraged me to explore innovative ideas freely, and yet steered my work in the ideal direction. Prof. Chattopadhyay has consistently inspired me to pursue excellence. His expertise and guidance have greatly contributed to my academic development and fostered my overall growth. He has ensured that I have access to all the resources necessary for my work.

Dr. Sanyal has shown me the true essence of hard work. His steadfast dedication to enhancing our research, immense patience with my constant inquiries, and positive perspective have been a source of inspiration throughout my doctoral journey. He has also ensured that I have access to any resources I may need. Both of my supervisors have always encouraged me to seize any opportunity that could benefit my career, and I consider myself fortunate to have worked under their supervision.

I am grateful to my Research Advisory Committee members for their valuable feedback, which helped shape my work. I extend my special thanks to Prof. Prasenjit Majumder, Professor at DAIICT, Gandhinagar, and Visiting Faculty at TCG CREST, India, for his guidance and support.

I would like to thank the former heads of the Department of Information Technology, Prof. Bhaskar Sardar, Prof. Uttam Kumar Roy, and Dr. Parama Bhaumik, for their support. I also thank Prof. Bibhas Chandra Dhara, the current Head of the Department, for ensuring a smooth research experience. I appreciate all the departmental faculty and administrative staff for their timely assistance.

I acknowledge the human annotators who contributed to the evaluation process; without their cooperation, this research would not have been possible. I am also grateful to all my collaborators for their valuable input. I would like to thank my colleagues and friends for their unwavering support, fruitful collaboration, and significant contributions throughout this journey.

I sincerely thank all my students for their dedication and inspiration, which have greatly contributed to my growth and motivation.

I am forever indebted to my parents for their unparalleled love and sacrifices in every aspect of my life. I am especially grateful to them for being constant friends and for supporting me in all my decisions. I extend my heartfelt thanks to all my family members for their unwavering encouragement.

Tohida Rehman

To
my Father and Mother.

“Brevity is the soul of wit.”

— William Shakespeare, Hamlet.

Abstract

Text summarization refers to the procedure of condensing the key ideas present in a single document or a group of related documents. The main benefit of text summarization is that it reduces the amount of time a reader needs to extract the main information in the document. This efficiency is particularly crucial in today's era of abundant data spanning fields such as journalism, scientific research publications, healthcare, finance, and many more. In every research field, there is a vast amount of textual information, with new research articles being published daily. A typical scientific publication begins with an abstract summarizing the entire paper. Recently, there has been an increasing focus on research highlights, which provide an additional bulleted summary that complements the traditional abstract by condensing the key findings of the research paper. These research highlights help a reader quickly understand the primary contributions of the paper. Abstractive summaries, which resemble those written by humans, are favored over extractive summaries, which simply select and rearrange portions of the original text.

The primary aim is to extract significant key findings from a research paper using methods that summarize information in a human-like manner. Evaluating the quality of the generated summary or research highlights is crucial for summarization systems. Thus, this research also emphasizes the metrics and methods used to assess summary quality. This thesis outlines the developed research framework on models, datasets, and evaluation techniques for scientific text summarization, emphasizing the challenges and the imperative to address and resolve these issues.

In this thesis, we begin by tackling the complex issue of designing automatic abstractive text summarization systems. Our initial focus is on extracting key information or research highlights within scientific publications. As we work on generating concise, coherent, and relevant summaries or research highlights from a broad spectrum of research papers, our objective is to navigate and address the related challenges.

To summarize, we address the key challenges of abstractive summarization systems for research publications: (i) the need for a suitable dataset to generate research highlights, as recently published papers often include not only an abstract but also a bulleted

list of key findings that aids readers in quickly understanding the main contributions of the paper; (ii) the limitations of traditional extractive summarization methods, underscoring the necessity for more advanced abstractive summaries; and (iii) the shortcomings of traditional automatic metrics in evaluating abstractive models to ensure the consistency of the summaries generated by these approaches.

In this thesis, our contributions are as follows. **Firstly**, we explored generating research highlights from scientific articles using deep learning techniques and analyzed the impact of different input embedding types on the model’s performance. Additionally, we proposed a new multidisciplinary dataset called *MixSub* for highlight generation for various domains of scientific papers. **Secondly**, we examined the impact of named entities on the generation of research highlights. **Thirdly**, we evaluated which metrics are most suitable for assessing scientific document summarization systems and analyze the factual consistency of the generated summaries at the entity level. **Fourthly**, we delved into the effectiveness of pre-trained language models (PLMs) and large language models (LLMs) in generating accurate and meaningful titles for research papers. **Fifthly**, we evaluated the quality of extremely short summaries generated by ChatGPT-3.5 and assess their effectiveness in comparison to traditional summaries through human annotation, presenting the ‘*SilverCSPicoSum*’ dataset as a key contribution.

Keywords— Natural Language Generation, Deep Learning, Abstractive Text Summarization, Pointer-Generator Network, Large Language Models, Pre-trained Language Models, Scientific Dataset, GloVe, ELMo, and SciBERT Word Embeddings, ROUGE Score, METEOR Score, MoverScore, BERTScore, SciBERTScore, Factual Consistency Measurement Metrics

Contents

1	Introduction	1
1.1	Background of Text Summarization	2
1.1.1	Automatic Text Summarization (ATS) System Classifications . .	2
1.1.2	Applications of Automatic Text Summarization (ATS) Systems .	5
1.1.3	Challenges of Automatic Text Summarization (ATS) Systems . .	6
1.2	Abstractive Summarization of Scientific Documents	7
1.2.1	Challenges and Scope of Abstractive Summarization of Scientific Documents	8
1.2.2	Problem Definition & Research Gap	9
1.3	Research Objectives	10
1.3.1	Generating Research Highlights from Research Papers	10
1.3.2	Study the Role of Named Entities in Research Highlight Generation	10
1.3.3	Analyzing Entity-level Factual Consistency in Abstractive Summary	10
1.3.4	Auto-generating Titles of Research Papers	11
1.3.5	A Dataset for Very Short Summaries on Scientific Documents . .	11
1.4	Contributions	11
1.4.1	Generation of Research Highlights with Deep Learning: Exploring the Impact of Embedding Types on Model's Performance	11
1.4.2	A New Multidisciplinary Dataset for Generating Research Highlights	12
1.4.3	Entity-Driven Insights: Named Entity Recognition-based Automatic Generation of Research Highlights	12
1.4.4	Hallucination Reduction in Long-Form Text Summarization and Research Highlight Extraction	12
1.4.5	Automated Title Generation for Research Papers Using Pre-Trained and Large Language Models	13
1.4.6	SilverCSPicoSum: A Dataset of Very Short Summaries Generated with ChatGPT-3.5	13
1.5	Organization of the Thesis	13

2	Related Work	16
2.1	Extractive Text Summarization	17
2.1.1	Advantages	17
2.1.2	Disadvantages	17
2.1.3	Early Works on Extractive Approaches	17
2.1.4	Comprehensive Summarization Approaches: Topic-Based, Statistical, and Conceptual	18
2.1.5	Machine Learning-Based Approaches	19
2.1.6	Graph-Based Approaches	20
2.1.7	Heuristic and Optimization-Based Approaches	21
2.2	Abstractive Text Summarization	22
2.2.1	Advantages	23
2.2.2	Disadvantages	23
2.2.3	Major Approaches in Abstractive Summarization	23
2.2.4	Leveraging Pre-Trained Language Models and Large Language Models in Abstractive Summarization	25
2.2.4.1	Advantages of PLMs / LLMs	26
2.2.4.2	Disdvantages of PLMs LLMs	26
2.3	Hybrid Approaches to Text Summarization	26
2.3.1	Advantages	27
2.3.2	Disadvantages	27
2.3.3	Significant Works on Hybrid Approaches	27
2.4	Approaches for Extracting Research Highlights and Summarizing Scientific Papers	27
2.5	Reducing Hallucinations and Ensuring Factual Accuracy in Text Summarization	29
2.6	Datasets for Text Summarization	30
2.6.1	News Article Datasets	30
2.6.2	Datasets for Scientific and Research Articles	31
2.6.3	Legal Article Datasets	32
2.6.4	Social Media Datasets	32
2.7	Evaluation Metrics for Text Summarization	32
2.7.1	ROUGE	33
2.7.2	METEOR	34
2.7.3	MoverScore	34
2.7.4	BERTScore	35
2.8	Summary	35

3	Generation of Research Highlights with Deep Learning: Exploring the Impact of Embedding Types on Model's Performance	37
3.1	Introduction	37
3.2	Background and Motivation	38
3.2.1	Background	38
3.2.2	Motivation	38
3.3	Challenges and Opportunities	39
3.3.1	Challenges	39
3.3.2	Opportunities	39
3.4	Different Text Representations Methods	39
3.4.1	Traditional Approaches for Text Representation	40
3.4.1.1	One-Hot Encoding	40
3.4.1.2	Bag-of-Words (BoW)	40
3.4.1.3	Term Frequency-Inverse Document Frequency (TF-IDF)	41
3.4.2	Neural Approaches for Text Representation	41
3.4.2.1	Pre-trained Word-Embeddings	41
3.4.2.2	Global Vectors for Word Representation(GloVe)	42
3.4.2.3	FastText	42
3.4.2.4	Embeddings from Language Models(ELMo)	42
3.4.2.5	Transformer-based Pre-trained Language Models	43
3.5	Main Contributions	44
3.6	Pointer-Generator Model with GloVe Word Embeddings and Coverage Mechanism	44
3.6.1	Methodology	45
3.6.2	Experimental Setup	45
3.6.2.1	Dataset Details	46
3.6.2.2	Data Pre-Processing	46
3.6.2.3	Implementation Details	46
3.6.3	Results	47
3.6.4	Case Studies	47
3.6.5	Summary of Findings	48
3.7	Pointer-Generator Model with ELMo Word Embeddings and Coverage Mechanism	49
3.7.1	Methodology	49
3.7.1.1	ELMo Pre-Trained Word Representations	50
3.7.1.2	Pointer-Generator Model with ELMo Embeddings and Coverage Mechanism	50
3.7.2	Experimental Setup	51

3.7.2.1	Dataset Details	51
3.7.2.2	Data Pre-Processing	51
3.7.2.3	Implementation Details	51
3.7.3	Results	52
3.7.4	Case Studies	52
3.7.5	Summary of Findings	54
3.8	Pointer-Generator Model with SciBERT Word Embeddings and Coverage Mechanism	57
3.8.1	Methodology	57
3.8.1.1	BERT and SciBERT	58
3.8.1.2	Pointer-Generator Model with SciBERT Embeddings	59
3.8.1.3	Pointer-Generator Model with SciBERT Embeddings and Coverage Mechanism	60
3.8.2	Experimental Setup	61
3.8.2.1	Dataset Details	61
3.8.2.2	Data Pre-Processing	61
3.8.2.3	Implementation Details	62
3.8.3	Results	62
3.8.3.1	<i>K</i> -Fold Cross-Validation	64
3.8.3.2	Comparison with Pre-Trained Models	64
3.8.3.3	Analysis of Energy Consumption	65
3.8.4	Case Studies	68
3.8.5	Summary of Findings	68
3.9	Comparison with Previous Works	74
3.10	Discussion	75
3.11	Summary	76
4	A New Multidisciplinary Dataset for Generating Research Highlights	77
4.1	Background and Motivation	77
4.2	Challenges and Opportunities	78
4.2.1	Challenges	78
4.2.2	Opportunities	79
4.3	Dataset Construction	79
4.4	Experimental Setup	81
4.4.1	Datasets	81
4.4.2	Data Pre-Processing	81
4.4.3	Implementation Details	81
4.5	Evaluation and Results on the MixSub Dataset	82

4.6	Case Studies on MixSub Dataset	85
4.7	Discussion	86
4.8	Summary	87
5	Entity-Driven Insights: Named Entity Recognition-Based Automatic Generation of Research Highlights	88
5.1	Introduction	88
5.2	Background and Motivation	89
5.2.1	Background	90
5.2.2	Motivation	91
5.3	Challenges and Opportunities	91
5.3.1	Challenges	91
5.3.2	Opportunities	92
5.4	Main Contributions	92
5.5	Methodology	93
5.5.1	NER-based Pointer-Generator Network	93
5.6	Experimental setup	94
5.6.1	Dataset Details	94
5.6.2	Data Processing	94
5.6.3	Implementation Details	95
5.7	Results	95
5.7.1	Comparison of Pointer-Generator type Models	95
5.7.2	Manual Evaluation	96
5.8	Case Studies	96
5.9	Discussion	100
5.10	Summary	100
6	Hallucination Reduction in Long-Form Text Summarization and Research Highlight Extraction	101
6.1	Introduction	101
6.2	Background and Motivation	102
6.2.1	Background	102
6.2.2	Motivation	103
6.3	Challenges and Opportunities	104
6.3.1	Challenges	104
6.3.2	Opportunities	104
6.4	Main Contributions	105
6.5	Methodology	105

6.5.1	Fine-tuning LED	106
6.5.2	Entity-based Data Filtering	106
6.5.3	Join sAlient ENtity and Summary Generation (JAENS)	106
6.6	Experimental Setup	107
6.6.1	Dataset Details	107
6.6.2	Data Processing	107
6.6.3	Implementation Details	108
6.6.4	Evaluation Metrics	109
6.7	Results	110
6.7.1	Results on PubMed Dataset	110
6.7.2	Results on CSPubSum Dataset	112
6.8	Case Studies	113
6.9	Discussions	118
6.10	Conclusion	119
7	Automated Title Generation for Research Papers Using Pre-Trained and Large Language Models	120
7.1	Background and Motivation	120
7.1.1	Background	121
7.1.2	Motivation	122
7.2	Challenges and Opportunities	122
7.2.1	Challenges	122
7.2.2	Opportunities	123
7.3	Main Contributions	123
7.4	Methodology	124
7.5	Experimental Setup	125
7.5.1	Dataset Construction	125
7.5.2	Datasets Used	125
7.5.3	Data Processing	126
7.5.4	Implementation Details	127
7.5.5	Evaluation Metrics	128
7.6	Results	129
7.6.1	Quantitative Comparison of Various Fine-Tuned Models	129
7.7	Case studies	132
7.8	Manual Evaluation	134
7.9	Demo	134
7.10	Discussion	135
7.11	Conclusion	136

8 SilverCSPicoSum: A Dataset of Very Short Summaries Generated with ChatGPT-3.5	137
8.1 Introduction	138
8.2 Background & Related Work and Motivation	138
8.2.1 Background & Related Work	138
8.2.2 Motivation	140
8.3 Challenges and Opportunities	141
8.3.1 Challenges	141
8.3.2 Opportunities	141
8.4 Main Contributions	141
8.5 Dataset construction	142
8.6 Quality Assessment for SilverCSPicoSum	143
8.6.1 Factual Consistency	143
8.6.2 Human annotation of SilverCSPicoSum	144
8.7 Methodology	144
8.8 Experimental setup	146
8.8.1 Dataset pre-processing	146
8.8.2 Implementation details	146
8.9 Results and Analysis	147
8.9.1 Manual evaluation	148
8.9.2 Analysis of energy consumption	150
8.10 Case Studies	151
8.11 Discussion	151
8.12 Conclusion	151
9 Conclusion and Future Scope	153
9.1 Summary	154
9.2 Future Scope	156
9.2.1 Multi-document Summarization Systems for Scientific Papers . .	156
9.2.2 Multi-lingual Research Highlight Generation for Scientific Papers	156
9.2.3 Metrics for Abstractive Summarization Systems for Scientific Doc- uments	157
References	158
10 List of Acronyms	180

List of Figures

1.1	Classifications of Automatic Text Summarization Systems	3
1.2	Architecture of both extractive and abstractive text summarization systems	4
1.3	Overview of Research Problems, Objectives, and Thesis Contributions. .	15
3.1	GloVe based encoder-decoder model for research highlight generation. .	45
3.2	Original abstract, author-written research highlights and research highlights generated by pointer-generator type models with GloVe embeddings. The meaning of the colors (e.g., green = correct) is explained in main text. Abstract taken from https://www.sciencedirect.com/science/article/abs/pii/S0168874X15000621	48
3.3	ELMo based encoder-decoder model for research highlight generation. .	49
3.4	The input consists only of the abstract of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without ELMo embeddings are shown. The input abstract and the author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S037722171500702X	54
3.5	The input consists only of the abstract of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without ELMo embeddings are shown. The input paper is at https://www.sciencedirect.com/science/article/pii/S0010482516300154	55
3.6	The input consists of the concatenation of the abstract, introduction, and conclusion of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without ELMo embeddings are shown. The input paper is at https://www.sciencedirect.com/science/article/pii/S0010482516300154	56
3.7	Proposed model: Pointer-generator network with coverage mechanism and SciBERT word embeddings.	58
3.8	Comparison of compute resources used by summarization models.	67

3.9	Input is only the abstract from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S0010482514001565	69
3.10	Input is only the conclusion from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S0010482514001565	70
3.11	Input is only the introduction from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S0010482514001565	71
3.12	Input is (abstract + conclusion) from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S0010482514001565	72
3.13	Input is (introduction + conclusion) from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from https://www.sciencedirect.com/science/article/pii/S0010482514001565	73
4.1	Subject-wise distribution of papers in MixSub dataset.	80
4.2	Input is only the abstract of an article from the MixSub dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights taken from https://www.sciencedirect.com/science/article/pii/S1567173920301292	86
5.1	Named entity recognition example, showcasing entities like organizations, people, locations, dates, monetary values, and percentages.	90
5.2	Proposed model: NER-based pointer-generator network with coverage mechanism.	94

5.3	Input is only an abstract from CSPubSum dataset. Highlights produced by the pointer-generator type models with and without NER are shown. Input and author-written research highlights taken https://www.sciencedirect.com/science/article/pii/S037722171300307X	97
5.4	The input consists of the concatenation of the abstract and conclusion of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without NER are shown. Input and author-written research highlights taken https://www.sciencedirect.com/science/article/pii/S0010448514001857	98
5.5	The input consists of the concatenation of the introduction and conclusion of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without NER are shown. Input and author-written research highlights taken https://www.sciencedirect.com/science/article/pii/S0010448514001821	99
6.1	Comparison of ground-truth and generated summary from the test set of the PubMed dataset. Hallucinations are highlighted. Input and ground-truth summary (abstract) taken from https://pubmed.ncbi.nlm.nih.gov/22629279/	115
6.2	Comparison of author-written and generated research highlights from the test set of the CSPubSum dataset. Hallucinations are highlighted. Input and author-written research highlights taken from https://www.sciencedirect.com/science/article/pii/S0010482514001681	116
6.3	Comparison of author-written and generated research highlights from the test set of the CSPubSum dataset. Hallucinations are highlighted. Input and author-written research highlights taken from https://www.sciencedirect.com/science/article/pii/S0010482514003266	117
7.1	Input is an abstract from CSPubSum dataset. Titles generated by the different models are shown. Paper taken from https://www.sciencedirect.com/science/article/abs/pii/S037722171500586X	132
7.2	Input is an abstract from LREC-COLING-2024 dataset. Titles generated by the different models are shown. Paper taken from https://aclanthology.org/2024.lrec-main.132/	133
7.3	Input is an abstract from LREC-COLING-2024 dataset. Titles generated by the different models are shown. Paper taken from https://aclanthology.org/2024.lrec-main.68/	134

7.4	Graphical user interface of our pre-trained language model-based title generation application.	135
8.1	Length statistics mean, min, max, and standard deviation (SD) for abstracts and pico summaries in the SilverCSPicoSum dataset, across Train, Validation, and Test sets.	143
8.2	Framework showing the SilverCSPicoSum dataset construction, model application, and evaluation metrics.	146
8.3	Comparison of compute resources used by summarization models.	150
8.4	Comparison of short summaries generated from an abstract by different models. Abstract taken from https://www.sciencedirect.com/science/article/pii/S026288561400047X	152

List of Tables

3.1	Evaluation of pointer-generator type models with GloVe embeddings: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore are based on abstracts as input from the CSPubSum dataset. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. All scores are presented as percentages (%).	47
3.2	Evaluation of pointer-generator type models with and without ELMo embeddings: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on different inputs from the CSPubSum dataset. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. All scores are presented as percentages (%).	52
3.3	Evaluation of pointer-generator type models with and without SciBERT embeddings: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on various inputs from CSPubSum dataset. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. All scores are presented as percentages (%).	63
3.4	K -fold cross-validation of the pointer-generator type models with and without SciBERT embeddings on the CSPubSum dataset. For comparison, the performance of the models with holdout validation are reproduced from Table 3.3.	64
3.5	Performance of fine-tuned versions of pre-trained models on the CSPubSum dataset using abstracts of the papers as the input. All metrics (ROUGE-1, ROUGE-2, ROUGE-L, METEOR, and BERTScore) are reported as F1-scores and presented as percentages (%). The highest scores are highlighted in bold.	65
3.6	Power consumption, compute expenditure, and CO ₂ emission statistics for summarization models.	67

3.7	Comparison of the performance of the our three proposed model with that of other approaches for CSPubSum dataset. F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on different inputs from the CSPubSum dataset.	74
4.1	Subject-wise URL count in MixSub dataset.	80
4.2	Comparison of Dataset Statistics: CSPubSum vs. MixSub.	81
4.3	Evaluation of pointer-generator type models with and without SciBERT embeddings: F1-scores for ROUGE, METEOR, MoverScore and BERTScore on MixSub dataset. The first row (where dataset is ‘Full MixSub’) indicates the performance when the models are trained on the whole MixSub training set and evaluated on the whole MixSub test set, without distinguishing between the subject categories of the papers. In the remaining part of the table, two cases are considered: Case 1: Trained on each subject-cluster of MixSub training set and evaluated on the corresponding test set; Case 2: Trained on the entire MixSub training set and evaluated on each subject-cluster of MixSub test set. All scores are presented as percentages (%).	84
5.1	Evaluation of pointer-generator type models with and without NER: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on various inputs from CSPubSum dataset. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script.	96
6.1	Evaluation of variants of LED fine-tuned models with and without filtering and JAENS technique: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on inputs that are the article body from the PubMed dataset. All scores are presented as percentages (%).	111
6.2	Evaluation of the variants of LED fine-tuned models with and without filtering and JAENS technique: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated summaries for the PubMed dataset. All scores in percentage (%).	112
6.3	Evaluation of variants of LED fine-tuned models with and without filtering and JAENS technique: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on inputs that are the abstract only from the CSPubSum dataset. All scores are presented as percentages (%).	113

6.4	Evaluation of the variants of LED fine-tuned models with and without filtering and JAENS technique: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated summaries for the CSPubSum dataset. All scores in percentage (%).	113
7.1	Some statistics of CSPubSum dataset and LREC-COLING-2024 dataset. #Max: Number of maximum words; #Avg: Number of average words; #Min: Number of minimum words. Similarly used for sentences.	126
7.2	Evaluation of all used models: F1-scores for ROUGE, METEOR are used for evaluating the both word-level matching and MoverScore, BERTScore, and SciBERTScore are used to compute the semantic fidelity of the generated titles for the CSPubSum dataset. All scores in percentage (%).	130
7.3	Evaluation of all used models: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated title for CSPubSum dataset. All scores in percentage (%).	130
7.4	Evaluation of all used models: F1-scores for ROUGE, METEOR, MoverScore, BERTScore and SciBERTScore for LREC-COLING-2024 dataset. All scores in percentage (%).	130
7.5	Evaluation of all used models: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated title for LREC-COLING-2024 dataset. All scores in percentage (%).	131
8.1	Comparison of SilverCSPicoSum with existing datasets in the scientific paper summarization domain. #Documents represents number of documents, the average word counts for input & summary, the input data source, the summary origin, and the domain. Abstract, Introduction, and Conclusion are abbreviated as Abs, Int, and Con.	142
8.2	Detailed breakdown of Named-Entity counts and Precision Scores in the SilverCSPicoSum dataset across different splits.	144
8.3	Evaluation of all models: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on the SilverCSPicoSum dataset using abstracts as input. All scores are presented as percentages (%).	148

8.4	Human evaluation score of the summary generated by the 3 fine-tuned models with respect to the silver standard pico summary from SilverC-SPicoSum dataset, based on Adequacy, Fluency, Coherence and Correctness.	149
8.5	Human evaluation scores for summaries generated by three fine-tuned models (BART-large, T5-base, LLaMA-3-8B) and one additional model (PGM + Coverage + SciBERT). Scores are based on the number of times each model-generated summary was chosen as the most or least preferred.	149

1

Introduction

The vast and rapidly expanding availability of online information across all domains has driven the Natural Language Processing (NLP) community to focus extensively on developing automatic text summarization systems. Due to the increasing volume of textual data across multiple domains – including news articles, academic papers, legal documents, biomedical texts, social media posts, blogs, and user reviews – Automatic Text Summarization (ATS) has become increasingly vital. The exponential growth of online content means that users often struggle to find, read, and comprehend all relevant information from search results. Manual summarization is time-consuming, labor-intensive, and costly, making it impractical for such a large volume of data. This has made the development of automated summarization systems not only urgent but essential. Since the 1950s, researchers have been dedicated to improving ATS techniques to address these challenges effectively.

In the research paper publishing domain, the volume of scientific papers is growing at an exponential rate [27], with reports indicating that the number of scientific articles approximately doubles every nine years [194]. This substantial increase in publications poses significant challenges for tracking new research, even within specialized sub-fields. Researchers often find it exceedingly difficult to stay updated with developments in their areas of interest. To address this, many publishers now request that authors provide a bulleted list of research highlights alongside the abstract and full text. This bulleted points of research highlights help readers quickly grasp the main contributions of the

paper. ATS involves the use of automated processes to generate concise summaries of longer texts. This process typically includes identifying and extracting the most significant information, ensuring that the resulting summary effectively represents the main ideas and key points of the original content.

The main aim of an ATS system is to condense a lengthy single document or similar multi-documents into a brief summary that conveys its essential ideas while avoiding unnecessary repetition. Such systems are designed to help users quickly understand the main content without needing to read the entire text. The summaries produced should be shorter than the original document and should effectively encapsulate its most important information, thereby saving users significant time and effort.

1.1 Background of Text Summarization

Automatic text summarization (ATS) is a technique that condenses lengthy documents, such as news articles, research papers, reports, social media news, blogs into concise summaries. The main advantage of ATS is its ability to reduce the time readers need to extract key information, thereby facilitating quicker comprehension. By enhancing semantic representation and data compression [100], ATS improves the efficiency of information retrieval systems and supports better management of large volumes of text [61].

This section explores the main aspects of automatic text summarization (ATS) and its increasing significance. It begins with a classification of different ATS systems in subsection 1.1.1, covers their various applications in subsection 1.1.2, and highlights the challenges they face in subsection 1.1.3 in an era of information overload.

1.1.1 Automatic Text Summarization (ATS) System Classifications

ATS systems can be classified based on several factors, including the number of input documents, the nature of the generated summary, the language of the summary, the summarization algorithm used, the summarization approach, the style and length of the generated summary, content of the summary, and the domain or dataset used for summarization [53]. As illustrated in Figure 1.1 automatic text summarization (ATS) systems can be categorized based on any of these criteria.

1. **Classification based on the input size:** The input size refers to how many source documents are utilized to generate the final summary. **Single-document summarization (SDS)** focuses on condensing the content of a single document while retaining its essential information. In contrast, **multi-document summarization (MDS)** produces a summary from multiple documents, with an

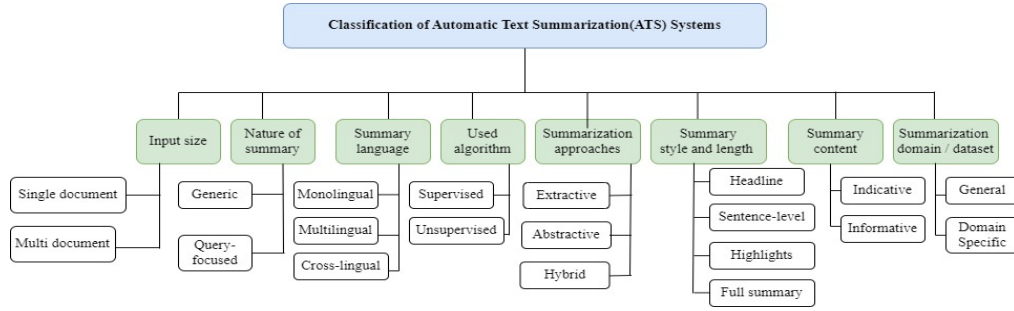


FIGURE 1.1: Classifications of Automatic Text Summarization Systems

emphasis on removing redundant information found across the set of documents.

2. **Classification based on nature of the generated summary:** Summaries can be categorized based on the nature of the generated summary as **generic** or **query-focused**. A generic summarizer extract important information from one or more documents to provide a general overview. On the other hand, query-based summarization works with documents found through a search query, focusing on information related to that query. While a generic summary gives a general sense of the document, a query-based summary emphasizes details relevant to the specific search and is sometimes called query-focused, topic-focused, or user-focused.
3. **Classification based on the language of the summary:** Based on the summary language, summarization systems can be classified into three types: **monolingual**, **multilingual**, and **cross-lingual**. Monolingual systems work with texts where the source and the summary are in the same language, such as summarizing an English article into an English summary. Multilingual systems process texts in multiple languages and produce summaries in those languages; for example, summarizing documents in English, Bengali, and Hindi into summaries in each of these languages. Cross-Lingual systems generate summaries in a different language from the source text, such as summarizing a Bengali article into an English summary.
4. **Classification based on the algorithm used for summarization:** Summarization systems can be categorized based on the algorithms they use, into **supervised** and **unsupervised** approaches. Supervised algorithms necessitate a training phase with pre-annotated data, which includes a reference summary. This approach involves manual annotation, which can be both labor-intensive and expensive. Conversely, unsupervised algorithms operate without requiring annotated data, so no reference summary is needed.

5. **Classification based on the approaches used for summarization:** Based on the used approaches for summarization, it can be called as **extractive** [92], **abstractive** [53] or **hybrid** types of summarization. Extractive methods generally copy whole or part of the sentences from the input source text and combine them into a summary, discarding the less important sentences from the input. Conversely, abstractive methods can generate novel words during summarization similar to how a human being does the job, that is, first reads the whole document, understands it, then summarizes by inducing new and appropriate words. The hybrid text summarization method combines extractive and abstractive techniques. Initially, it identifies key sentences from the input text using an extractive approach, and then applies abstractive methods to generate the final summary. See Figure 1.2.

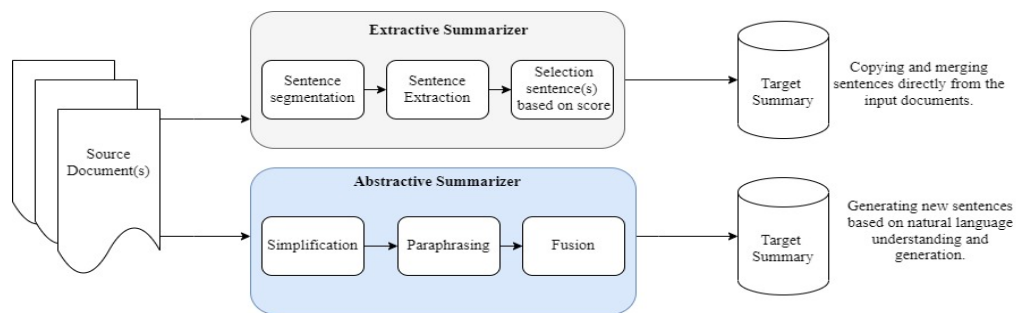


FIGURE 1.2: Architecture of both extractive and abstractive text summarization systems

6. **Classification based on the style and length of the generated summary:** Based on the style and length of the generated summary, it can be classified as **headline generation**, **sentence-level summary**, **highlights generation**, or **full summary**. Headline generation produces a brief headline, sentence-level summarization yields a single sentence, highlights summarization provides a concise, bulleted-point overview, and full summary generation adapts to the required length or compression ratio. For each category, the summary may be extractive, abstractive, or hybrid depending on the approach used.
7. **Classification based on the content of the summary:** Summaries can be classified as **indicative** or **informative** based on their content [53]. An informative summary offers a detailed overview of the main points, key ideas, and important details of the original text, including aspects like research questions, methodology, findings, and conclusions. In contrast, an indicative summary provides a general idea of the topics or themes discussed, summarizing the content broadly without delving into specifics or detailed results.

8. **Classification based on the summarization domain / dataset used:** Summaries can be categorized based on the dataset used into two primary types: **general** and **domain-specific**. General summarization systems are capable of managing a wide variety of document types across different fields. They generate summaries that are broadly applicable but may not capture the specific nuances of any particular field. On the other hand, domain-specific summarization systems are designed to produce detailed summaries tailored to particular areas of expertise. They apply specialized knowledge and terminology pertinent to specific domains, such as medical, legal, scientific, news, or social media contexts, to create more accurate and contextually relevant summaries.

1.1.2 Applications of Automatic Text Summarization (ATS) Systems

Automatic Text Summarization (ATS) is extensively used in text mining and analytics applications, including information retrieval, information extraction, and question answering. When combined with information retrieval techniques, ATS enhances the performance of search engines by providing more relevant and concise search results. ATS systems have diverse applications across various fields. Here are some key areas where ATS is commonly used:

1. **Academic research papers summarization:** ATS systems for scientific papers simplify complex research documents by summarizing key findings and essential details [109]. Consequently, they can provide highlights from research papers that not only help researchers precisely and quickly identify the contributions of a paper but also enhance the discoverability of the article via search engines[43]. Currently, many publishers are requesting the submission of bulleted points of research highlights along with the abstract and full paper. This can distill large volumes of information into concise, informative summaries, making it easier for researchers to identify relevant studies and insights.
2. **News articles summarization : Newsblaster** is an example of a news summarizer designed to help users discover relevant news [119]. It operates by automatically collecting, clustering, categorizing, and summarizing news from various sources such as CNN and Reuters on a daily basis.
3. **Legal documents analysis:** ATS systems are employed to distill key information from legal documents like contracts, case law, and briefs [78].
4. **Biomedical documents summarization:** ATS systems for biomedical documents [113] extract and summarize key information, such as study findings, treatment methods, and patient details [109, 15].

5. **Social media summarization:** Social media information summarization involves capturing key themes, trends, and sentiments from tweets and posts[198].
6. **Books/Story/Novel summarization:** Summarizing books, stories expose main points, key characters, and central themes [123, 83].
7. **Email summarization:** ATS systems highlight key topics, action items, and deadlines [192].
8. **Customer feedback summarization:** ATS systems highlight common themes, overall sentiments, and actionable insights [36].
9. **Technical support documents summarization:** ATS systems find out common issues, solutions, and troubleshooting steps [218].

1.1.3 Challenges of Automatic Text Summarization (ATS) Systems

To achieve an accurate summary comparable to a human-written one, several challenges must be addressed. These challenges arise from the complexity of condensing information while preserving key details and ensuring the summary remains coherent and informative. The specific challenges faced are outlined below:

1. **Factual consistency and hallucinations:** One of the key issues in abstractive summarization is ensuring factual consistency between the generated summary and the original text. Abstractive models, especially those using deep learning techniques, can sometimes generate information that wasn't present in the source text, leading to hallucinations, which means misinterpreting the information. This can be problematic in the fields like medicine, law, scientific paper analysis and many others where accuracy is very important and critical.
2. **Handling long documents:** Effectively summarizing long documents is a major challenge. Many summarization models find it difficult to handle lengthy texts because of memory and computational limits. Summarizing long texts often requires condensing information while retaining key details, which is challenging to achieve consistently. Even though models like Longformer [22], which can process up to 4,096 tokens, have improved this, they still face limitations.
3. **Context understanding and coherence:** Summarization systems must understand the context and maintain the coherence throughout the summary. This

requires a deep understanding of the text, including relationships between different sections, which is difficult for many models. Coherent summaries should flow logically and cover the main points without abrupt shifts in topics.

4. **Domain specific knowledge:** Domain specific summarization systems, need to have a deep understanding of the specialized language and concepts used in that field. Training models with domain-specific knowledge is very challenging due to the need for a high-quality large datasets specific to that domain. Hence, general models struggle to generate accurate and relevant summaries in any specialized fields.
5. **Handling multilingual and Cross-language summarization:** With the growing need to summarize texts in multiple languages or even translate and summarize simultaneously, the challenge of multilingual and cross-language summarization becomes apparent. Models need to handle linguistic nuances, cultural context, and differences in syntax across languages.
6. **Evaluation metrics:** Assessing the quality of summaries is challenging because traditional metrics, such as ROUGE [102] and METEOR [17], primarily measure word overlap between the generated summary and the reference summary, rather than content accuracy or readability. In contrast, MoverScore [227] and BERTScore [224] aim to evaluate the semantic similarity between the generated and reference summaries by using latent text representations (embeddings). There is a need for improved evaluation methods that consider factors such as factual accuracy, readability, informativeness, and the degree of abstraction in abstractive summaries.

1.2 Abstractive Summarization of Scientific Documents

It has already been mentioned that the rapid growth in the number of scientific papers is overwhelming, with reports showing that the volume of articles roughly doubles every nine years [194].

For a researcher, keeping track of any research field is extremely difficult even in a narrow sub-field. Nowadays, many publishers request authors to provide a bulleted list of research highlights along with the abstract and the full text. It can help the reader to quickly grasp the main contributions of the paper. Scientific papers are typically more comprehensive and organized differently from news articles. The main aim of this work is to develop a system to produce clear and impactful research highlights or titles from scientific papers, effectively presenting the key contributions and insights. The necessity can be analyzed from two different angles.

- **Managing huge publications:** Firstly, the rapid increase in the number of scientific publications presents a challenge for researchers to keep up with the latest developments. Effective summarization techniques help by distilling large volumes of research publications into concise summaries, enabling researchers to manage and understand extensive amounts of information more efficiently.
- **Improving research efficiency:** Secondly, abstractive summarization provides clear and concise research highlights, enabling researchers to quickly grasp key points and details for easier review and understanding.

1.2.1 Challenges and Scope of Abstractive Summarization of Scientific Documents

We now highlights some challenges associated with abstractive summarization of scientific documents. Abstractive summarization of scientific documents faces several challenges. First, scientific papers often use complex terms and specialized language that can make it hard to create clear summaries. These documents are packed with detailed information, and it can be difficult to reduce papers content without losing important details. Abstractive summarization models also need to understand the context and specifications of the research to produce meaningful summaries. It is crucial that these summaries accurately reflect the original work. Additionally, traditional metrics such as ROUGE [102], METEOR [17], MoverScore [227] and BERTScore [224] for evaluating summaries might not be enough. There is a need for better evaluation method that consider aspects such as accuracy, readability, informativeness, and the level of abstraction in summaries.

The scope of this research encompasses several critical aspects of abstractive summarization for scientific documents. **Firstly**, methods are needed to generate clear and insightful summaries or research highlights of scientific papers. This involves applying advanced machine learning techniques or deep learning techniques to extract complex and detailed information into concise summary or research highlights. **Secondly**, scientific documents often contain specialized jargon and complex terminology. The scope includes developing approaches to effectively manage and interpret this specialized language or semantics meaning to ensure that summaries remain accurate and comprehensible. **Thirdly**, How identification and use of scientific named entities can enhance the relevance and precision of generated research highlights or summaries may be explored. **Finally**, another critical area is the development of robust evaluation metrics tailored to abstractive summarization systems. This involves adapting methods to assess the quality of summaries in terms of factual accuracy, readability, informativeness, and coherence, and how they can be match with human-written summaries.

Based on the preceding discussions, the thesis outlines problem definition & research gap, its research objectives, followed by a summary of the associated contributions.

1.2.2 Problem Definition & Research Gap

After conducting a thorough literature survey, we found that most existing models for highlights generation are primarily extractive, and the available human-annotated datasets are predominantly domain-specific, particularly in the computer science domain. This highlights the need to develop abstractive summarization systems capable of producing concise research highlights across a broader range of scientific domains.

While pre-trained language models (PLMs) and large language models (LLMs) have demonstrated remarkable capabilities across various Natural Language Processing (NLP) applications, the generation of silver-standard datasets for deep learning models—especially those requiring large amounts of data remains relatively under explored.

Additionally, current evaluation metrics do not sufficiently ensure factual consistency when summarizing scientific documents, leaving room for improvement in this area.

Abstractive summarization of scientific documents faces several challenges. Scientific papers often use complex terms and specialized language, making it difficult to create clear and compact summaries. Additionally, these documents are rich in detail, so reducing their content without losing crucial information is challenging. Abstractive summarization models must also grasp the context and specifics of the research to generate meaningful summaries.

1. **Concise summarization of research publications:** Our approach to develop abstractive summarization systems to generate clear and impactful research highlights, appropriate titles, or key contributions from scientific papers. *Research highlights* consist of a list of points summarizing the main findings of the paper, are typically written by the author along with the abstract. Research highlights can be considered a summary not only of the paper but also of the abstract. Automatically generating research highlights helps authors in writing their key contributions, as many publishers are requesting submission of the full paper along with research highlights. Similarly, by providing a concise summary of the most important aspects of the research, the highlights help readers quickly assess the relevance of the paper and determine whether it is worth reading in full.
2. **Evaluating the quality of AI-generated abstractive summaries:** Abstractive summarization involves generating new, coherent phrases that capture the essence of the original content. It is crucial that these summaries accurately reflect the original work. Traditional metrics such as ROUGE [102], METEOR

[17], MoverScore [227], and BERTScore [224] may not fully capture aspects of summary quality. There is a need for better evaluation methods that consider accuracy, readability, informativeness, and the level of abstraction in summaries.

3. **Mitigating the problem of manual dataset annotation:** Creating high-quality annotated datasets for training deep learning models is a time-consuming and labor-intensive process. Developing methods to automate or streamline this process is crucial for advancing summarization technologies.

1.3 Research Objectives

Considering the background, motivation, research gap, problem statement, and challenges related to automatic text summarization – particularly in the area of abstractive summarization of scientific documents – this thesis outlines its primary research objectives. In summary, the objectives of this thesis are listed below.

1.3.1 Generating Research Highlights from Research Papers

After identifying the research gap in the domain of abstractive summarization of scientific documents, the first objective of this thesis is to develop an abstractive summarization model for generating research highlights from scientific papers. An additional objective is to propose a new dataset for highlight generation for various domains of scientific papers.

1.3.2 Study the Role of Named Entities in Research Highlight Generation

In scientific document summarization, it is crucial to ensure that multi-word tokens are accurately grouped together to generate accurate and meaningful summaries. To achieve this, we need to develop a mechanism that effectively combines Named Entity Recognition (NER) with advanced deep learning techniques for abstractive summarization or highlights generation from research papers, ensuring that entity phrases remain intact in the generated summary.

1.3.3 Analyzing Entity-level Factual Consistency in Abstractive Summary

When summarizing lengthy texts or scientific documents, a common issue is the occurrence of hallucinations. This is particularly challenging because scientific papers are

generally longer than news stories and have a different discourse structure. Hallucinations arise when a summary contains inaccuracies or adds information not present in the original document. Additionally, deep learning models used for generating abstractive summaries may introduce named entities that were not part of the source text, a challenge referred to as the entity hallucination problem [130]. Addressing this issue involves conducting analyses to identify and understand the presence of hallucinations, with techniques for mitigation being an important aspect to explore.

1.3.4 Auto-generating Titles of Research Papers

A research paper’s title is essential in effectively summarizing the main theme and main findings in a concise way. However, coming up with an accurate and right title can be a arduous task and time-intensive process for authors. Therefore, an automated system for generating titles could offer valuable assistance and enhance efficiency for novice researchers.

1.3.5 A Dataset for Very Short Summaries on Scientific Documents

In response to the challenges of limited datasets and the high cost of creating large-scale human-annotated datasets, there is a need for very short summaries of scientific documents, possibly, using deep learning models. Most scientific papers do not provide open access to their full texts – this situation has motivated us to construct extreme pico-summary datasets from research papers.

1.4 Contributions

In light of the above research objectives, the major contributions of this thesis are outlined below.

1.4.1 Generation of Research Highlights with Deep Learning: Exploring the Impact of Embedding Types on Model’s Performance

In this work, we introduced three deep learning models designed to generate research highlights from academic papers. Additionally, it evaluates how various types of word embeddings influence the models’ performance. The number of scientific publications approximately doubles every nine years [194], posing a challenge for researchers to stay updated even within their own areas of expertise. A recent trend involves providing **research highlights** – a bulleted summary of the paper’s main contributions – alongside the abstract and other sections of the main text. These highlights offer a more concise and focused overview of the *key findings*, making them easier to read on mobile

devices compared to traditional abstracts, which often include extensive background information.

For this purpose, to generate research highlights from a paper, we have proposed an abstractive summarization model based on a pointer-generator network extended with a coverage mechanism [174].

Our first work [163] in this thread uses pre-trained GloVe embeddings [141] to represent the tokens of the input document.

In the second work [162] in this thread, we replaced the Global Vectors for Word Representation (GloVe) embeddings with pre-trained ELMo contextual embeddings [143]. Embeddings from Language Models (ELMo), being contextually sensitive, distinguishes homonyms by assigning distinct vectors even when these words share the same spelling.

In the third work [164] in this thread, we passed the source document through SciBERT [21] and input the resulting token embeddings into the coverage-augmented pointer-generator network.

1.4.2 A New Multidisciplinary Dataset for Generating Research Highlights

In this work, we proposed a new dataset called *MixSub*, which contains research articles from multiple domains [164]. To prepare *MixSub*, we crawled the ScienceDirect¹ website and curated articles published in various journals in 2020. We removed articles that did not contain research highlights, resulting in a final collection of 19,785 articles with author-written research highlights.

1.4.3 Entity-Driven Insights: Named Entity Recognition-based Automatic Generation of Research Highlights

In this work, we have introduced a novel approach that integrates named entity recognition (NER) with pointer-generator networks incorporating a coverage mechanism to automatically generate research highlights from a research paper. To our knowledge, this is the first research to combine NER with pointer-generator networks and a coverage mechanism [174] for the purpose of generating research highlights.

1.4.4 Hallucination Reduction in Long-Form Text Summarization and Research Highlight Extraction

In this work, we have used the Longformer Encoder-Decoder (LED) model to generate summaries and research highlights from scientific papers, incorporating data filtering

¹<https://www.sciencedirect.com/>

and JAENS to evaluate their effects on factual consistency. We assessed consistency using new metrics, such as precision-source and F1-target [130], in addition to traditional metrics like ROUGE, METEOR, and BERTScore.

1.4.5 Automated Title Generation for Research Papers Using Pre-Trained and Large Language Models

In this work, we have automated title generation from research paper abstracts by fine-tuning pre-trained and large language models. We compare their performance with ChatGPT in a zero-shot setting. Titles are evaluated using ROUGE, METEOR, MoverScore, BERTScore, and SciBERTScore metrics, and their factual accuracy is assessed at the entity level using precision-source and F1-target metrics from [130]. We also conduct a manual evaluation of a subset of titles.

1.4.6 SilverCSPicoSum: A Dataset of Very Short Summaries Generated with ChatGPT-3.5

In this work, we introduce the *SilverCSPubSum* dataset, comprising ScienceDirect papers with GPT-3.5-generated pico summaries, which reduces human annotation costs and time. We fine-tuned several pre-trained models, including T5, BART, Pegasus, ProphetNet, and LLaMA-2/3 with LoRA, and compared their performance with a pointer-generator network using SciBERT embeddings and a coverage mechanism. Performance was assessed using ROUGE, METEOR, MoverScore, and BERTScore metrics, supported by additional human assessments. We used the *Precision-source* metric to assess factual consistency, showing high precision and minimal hallucination. Human evaluations confirmed the summaries adequacy, fluency, coherence, and correctness.

1.5 Organization of the Thesis

In this section, the organization and content of the following chapters are briefly described.

Chapter 1 introduces the research background and motivation, followed by a detailed outline of the thesis objectives. Additionally, a flowchart is included to visually represent the mapping of research problems, objectives, and the contributions of the thesis.

Chapter 2 presents an extensive discussion of the existing literature in the field of automatic text summarization systems. In doing so, we discuss the different datasets,

approaches, and evaluation modalities proposed in such works and identify the limitations and future scopes.

Chapter 3 presents *Generation of Research Highlights with Deep Learning: Exploring the Impact of Embedding Types*, introduces an abstractive approach to generating highlights from research papers, and assesses the impact of different word embeddings on the model's performance.

Chapter 4 describes *A New Multidisciplinary Dataset for Generating Research Highlights*, which introduces a new dataset called *MixSub* from multiple domains for generating highlights from research papers.

Chapter 5 presents *Entity-Driven Insights: Named Entity Recognition-Based Automatic Generation of Research Highlights*, which augments named entity recognition with a deep learning model to accurately generate research highlights from research papers.

Chapter 6 presents *Hallucination Reduction in Long-Form Text Summarization and Research Highlight Extraction*, addressing the evaluation of abstractive summarization systems by analyzing the factual consistency of the generated summaries using both new and common metrics, with a focus on overlapping, semantic, contextual embeddings, and cosine similarity.

Chapter 7 presents *Automated Title Generation for Research Papers Using Pre-Trained models and Large Language Models*, which assesses the automation of research paper title generation from abstracts using pre-trained language models and large language models.

Chapter 8 presents *SilverCSPicoSum: A Dataset of Very Short Summaries Generated with ChatGPT-3.5*. It introduces the *SilverCSPicoSum* dataset, which is created from ScienceDirect papers and leverages GPT-3.5 to generate pico summaries. This approach addresses the challenge of obtaining large annotated datasets by reducing the cost and time associated with manual annotation.

Chapter 9 finally concludes the thesis by summarizing the findings from the previous chapters and outlining potential directions for future research in these areas.

The Figure 1.3 shows the overview of research problems, objectives and thesis contribution.

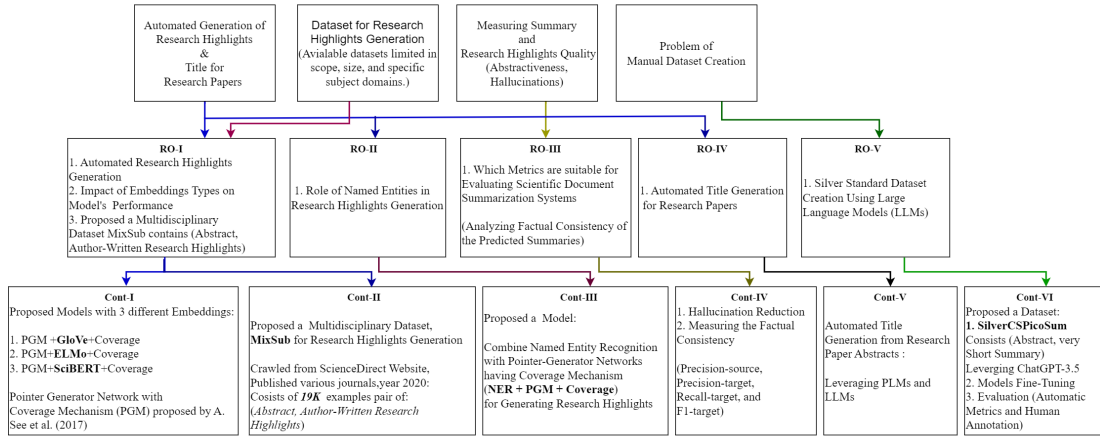


FIGURE 1.3: Overview of Research Problems, Objectives, and Thesis Contributions.

2

Related Work

This chapter provides a comprehensive review of state-of-the-art literature in the field of automatic text summarization, with a particular focus on abstractive summarization of scientific documents. The aim is to help researchers stay up-to-date with the latest findings in their domain. By analyzing the research advancements related to the concepts discussed in Chapter 1, this review also identifies opportunities for further exploration and establishes a foundation for the objectives and contributions presented in this thesis.

While the primary objective of this chapter is to provide an overview of the preliminary stages of the following chapters, we also try to answer the following Research Question (RQ) by studying the *state-of-the-art* literature:

RQ1 : What are the most effective algorithms for extractive text summarization across different text genres?

RQ2 : What strategies can improve the training of neural models and hybrid approaches to generate more coherent and contextually accurate abstractive summaries across various text genres?

RQ3 : How do extractive and abstractive approaches compare in their ability to capture the key contributions or generate research highlights of scientific papers?

RQ4 : What techniques can reduce hallucinations and ensuring factual accuracy in text summarization?

RQ5 : How do existing datasets and evaluation metrics influence the development and evaluation of text summarization models?

Section 2.1 will provide the answer of **RQ1**. Sections 2.2 and 2.3 will aim at deriving the answers to **RQ2**. Section 2.4 will provide answer to **RQ3**. Sections 2.5, 2.6 and 2.7 will provide answers to **RQ4–RQ5**.

2.1 Extractive Text Summarization

Various significant methods used in extractive text summarization are discussed in this section.

2.1.1 Advantages

The extractive approach is quicker and more straightforward than the abstractive method. It offers higher accuracy by directly selecting sentences or part of the sentences from the original text, only ensures that the summary uses the exact terminology from the source.

2.1.2 Disadvantages

The extractive summarization method often falls short of human-generated summaries due to the problems such as redundancy, excessively long sentences, and conflicting temporal references. Additionally, it may lack semantic coherence and fail to address important details effectively.

2.1.3 Early Works on Extractive Approaches

Automatic text summarization has a long history, evolving from early rule-based approaches to the sophisticated deep learning models used today. Its development reflects advancements in natural language processing and machine learning technologies. As far back as 1958, Luhn et al. [112] pioneered an extractive summarization technique that selects sentences based on the frequency of significant words, excluding common words, to summarize technical papers and magazine articles. The position of a sentence within a document is frequently used as a feature in document summarization [20]. In that research found that 85% of the theme sentences selected from the first sentences of the paragraph and 7% as the last sentence of a paragraphs. Edmundson et al.[52] proposed a method for automatically summarizing texts that assigns a score to each sentence based on four features including sentence position, word frequency, document skeleton, and cue words. Gerald DeJong et al. [49] proposed a knowledge-based summarization

approach called FRUMP (Fast Reading Understanding and Memory Program). This system achieved success in the unconstrained domain of news articles by effectively understanding and summarizing unseen stories.

Extractive summarization approaches can be broadly categorized into concept-based, statistical-based, topic-based, graph-based, semantic-based, machine learning-based, and deep learning-based methods [53].

2.1.4 Comprehensive Summarization Approaches: Topic-Based, Statistical, and Conceptual

TF-IDF, a statistical method that weights terms based on their frequency within a document and across a collection, has significantly influenced information retrieval and text summarization techniques [180]. A prototype intelligent information retrieval system, SCISOR (System for Conceptual Information Summarization, Organization, and Retrieval), was developed to implement a text processing strategy that can handle unknown words and gaps in linguistic knowledge while automatically acquiring lexical information from texts [157]. The paper advances text summarization by introducing methods for automatic text analysis, theme generation, and summary creation, along with evaluating their effectiveness [170]. Brandow et al. [28] introduced a technique for selecting words using $TF * IDF$ (*term frequency * inverse document frequency*), incorporating sentence-based features such as key terms, location, anaphora, and abstract length. Sentences lacking key terms are inserted between the chosen sentences. To enhance automatic indexing and retrieval, and to improve the relevance detection of documents, Latent Semantic Analysis (LSA) was initially introduced by Deerwester et al. [48]. A new summarization approach that generate coherent, style-specific summaries shorter than a single sentence, using statistical models for term selection and ordering, and can generate summaries that do not necessarily include words from the original text; it also demonstrates the capability to translate and summarize Japanese documents into English headlines [18, 208]. Steinberger et al. [181] demonstrated that LSA can be applied to identify and extract sentences that best represent each latent concept for text summarization. Additionally, Wang et al. [201] have enhanced this process by utilizing more interpretable concepts from HowNet and utilized concepts as features rather than individual words. Mashechkin et al. [117] proposed a summarization technique that used LSA for identifying key sentences, and they introduce a novel approach using non-negative matrix factorization, achieving superior performance on the DUC 2001 and 2002 datasets. Statistical techniques prove to be both adaptable and effective in the varied field of text summarization by their capability to extract

key information through measurable metrics [23]. Despite lacking the semantic complexity of newer techniques, statistical methods are still valuable due to their simplicity and efficiency. Statistical models and algorithms may have difficulty handling intricate language structures. They remain relevant in scenarios where computational power is limited or when a quick and straightforward summary is needed [9].

2.1.5 Machine Learning-Based Approaches

Statistical methods demand fewer computational resources, such as processing power and memory, and do not require additional linguistic expertise. However, they may occasionally fail to include significant sentences in the generated summary[53]. Kupiec et al. [92] introduced a first trainable extractive summarization model for scientific articles. Their method employed a Naive Bayes Classifier (NB) to categorize sentences as either part of the summary or not, based on various features like sentence length ($|s| > 5$), presence of uppercase words (except common acronyms) and other features (sentence position, word frequency, document skeleton, and cue words) from Edmundson et al. [52]. McKeown and Radev et al. [118] introduced “SUMMONS: Semantic Unification-based Multifaceted Multimedia Open Network Synthesis”, a knowledge-based system that utilizes information extraction methods akin to the MUC template from the Message Understanding Conference. This system uses slot fillers to generate text, resulting in detailed and cohesive summaries of multimedia content. This represents one of the *earliest works* in multi-document summarization, utilizing a knowledge-based approach. Aone et al. [13] integrated a Naive Bayes Classifier technique with more advanced features in their system, *DimSum*. This system utilized features such as term frequency (TF) and inverse document frequency (IDF) to identify key terms. Lin et al. [103] proposed and evaluated the “Optimal Position Policy (OPP)”, a strategy that ranks sentence positions to effectively identify key sentences within texts. This method, tailored to specific genres, was shown to be effective in applications like information retrieval and summarization. Later, Lin et al. [101] advanced sentence extraction by employing *decision trees* rather than a NB. Their study evaluates the heuristics for generating summaries from newspaper articles, investigates how topic prominence and query types influence performance, and explores the use of machine learning for combining heuristics. They also discuss the “SUMMARIST multilingual text summarization system”. Conroy and O’Leary applied a *hidden Markov model (HMM)* for sentence extraction, incorporating local dependencies and features like sentence position, length, and term likelihood, achieving better alignment with human summaries than previous methods [44].

Existing approaches to summarization have often assumed feature independence.

Osborne [137] addressed this by employing log-linear models and demonstrated that these models, particularly with an optimized prior, yield better sentence extracts compared to NB based models. His research highlighted that maximum entropy classifiers can significantly surpass NB, especially when utilizing highly informative features. Nenkova et al. [133] analyzed that a robust baseline for summarization, which involved using the initial sentences of a newswire article, finding that this approach effectively captures the key information due to the journalistic practice of highlighting important content early. *NetSum*, a neural network-based summarization method utilizing features from news query logs and Wikipedia entities, significantly outperforms standard baselines in ROUGE-1 metrics on CNN.com documents [186]. For ranking each sentence, the *RankNet* algorithm was used [31]. Neural network architectures like recurrent neural networks (RNNs) and transformers [195] allow machine learning models to understand long-range dependencies and semantic connections in the text, leading to the generation of more coherent and contextually appropriate summaries [37, 209]. However, machine learning methods present challenges, such as the necessity for vast amounts of labeled data and substantial computational power to train intricate models [196]. This work introduces a new summarization method employing trainable machine learning algorithms and features extracted from the text—both statistical and linguistic—and evaluates its performance against baseline methods using established text datasets [135].

Although machine learning methods face challenges such as the need for substantial labeled data and high computational resources for model training, they continue to advance text summarization. These approaches offer promising opportunities for generating high-quality summaries that address the evolving requirements of users in a data-centric world [207].

2.1.6 Graph-Based Approaches

Salton et al. [171] proposed a domain-independent method for automatic text summarization. They proposed graph-based techniques that involve breaking the text into components and using semantic hyperlinks between paragraphs with significant lexical similarity. This approach reorganizes the text based on connectivity, with highly interconnected paragraphs deemed more likely to contain key information relevant to the article’s topic. Mani and Bloedorn [114] proposed a summarization framework that utilized graph-based techniques to assess similarities and differences between documents. In their approach, content is represented as entities (nodes) and relationships (edges) within a graph. They identify key regions of interest by analyzing the graph’s structure, rather than extracting sentences directly. *LexRank*, a graph-based algorithm for extractive summarization approach proposed by Erkan and Radev et al.

[54] to calculate the importance of sentences based on their sentence similarity, and centrality as salience properties. It is an unsupervised approach. Radev et al. [151] developed *MEAD*, a summarization tool for multiple documents that leverages cluster centroids from topic detection and new evaluation methods. *MEAD*'s notable aspect is its use of cluster centroids, which identify key words central to all articles within a cluster. The approach encompasses position-based, centroid-based, largest common subsequence, and keyword-based methods. Mihalcea & Tarau et al. [124] introduced a new method showing that graph-based algorithms can independently and effectively summarize documents across various languages. This approach contrasts with earlier studies [54], which either concentrated on single-document English summaries or integrated graph-based methods with other techniques, thereby limiting the ability to evaluate the graph algorithms' effectiveness on their own. Rank sentences using established algorithms such as *HITS* [87] or *PageRank* [29]. *iSpreadRank* is a novel method for extracting sentences, utilizing graph-based ranking strategies and utilizing activation theory to assess and rank the importance of sentences for effective summarization [216]. Antiqueira et al. [12] introduced extractive summarizers that utilized complex network metrics for sentence ranking, demonstrating effectiveness comparable to advanced methods while relying on minimal text preprocessing. Ye, Chua, and Lu et al. [215] proposed a method for generating summaries using Wikipedia components, such as infoboxes and article sentences, and integrate these with concepts and non-textual features. Their model, tested on TREC-QA data, shows enhanced summarization and definition answering performance. Miao and Li et al.'s WikiSummarizer system enhances summarization by integrating Wikipedia concepts into sentence representations and achieved competitive results compared to baseline methods in the TAC 2010 evaluation [122]. Sankarasubramaniam et al. [173] proposed an innovative summarization technique integrating Wikipedia with graph-based ranking. They construct a bipartite graph linking sentences and concepts, and iteratively rank sentences to generate summaries. The method supports real-time incremental summarization and can be tailored to user interests and queries. Their approach, evaluated using ROUGE metrics [102] and user feedback, demonstrates significant improvements in summary quality.

2.1.7 Heuristic and Optimization-Based Approaches

Barzilay and Elhadad et al. [19] proposed a summarization technique that uses detailed linguistic analysis of lexical chains—sequences of related words across varying distances in a text to identify and extract important sentences, enhancing the coherence and relevance of the summaries. Marcu et al. [115] proposed a distinctive summarization approach that moves beyond the assumption of a linear sentence sequence by applying

discourse-based heuristics and incorporating Rhetorical Structure Theory (RST). Further elaborates on a rhetorical parser that generate a discourse tree to better represent the text’s hierarchical structure. Carbonell and Goldstein et al. [35] introduced the Maximal Marginal Relevance (MMR) criterion, designed to minimize redundancy and maintain query relevance, achieving notably better results in producing non-redundant multi-document summaries than conventional techniques. Silber & McCoy et al. [177] developed a linear-time algorithm for lexical chain computation, improving efficiency in ATS. This advancement makes lexical chains a viable intermediate representation for summarization processes. The work also includes a novel evaluation method for lexical chains, addressing the computational challenges of earlier methods. The study presents two methods for text summarization: the Modified Corpus-Based Approach (MCBA) using *genetic algorithms* for feature weighting, and the latent semantic analysis (LSA) combined with text relationship mapping (TRM) approach [217]. Alguliev et al. [6] proposed an unsupervised summarization approach that frames summarization as an optimization challenges, focusing on extracting crucial and unique sentences. Their method, evaluated using the DUC2005 and DUC2007 datasets, outperforms existing baseline systems. Alguliyev et al. [7] developed *COSUM*, a text summarization model that utilizes clustering and optimization techniques to improve coverage, diversity, and readability, achieving superior performance compared to 14 state-of-the-art methods in ROUGE [102] evaluations. Meena et al. [121] review the use of genetic algorithms in extractive text summarization, highlighting their evolution and effectiveness. They also propose an improved feature set for the fitness function and address challenges related to computational time and cost, stressing the importance of optimizing the number of iterations in these algorithms [120]. A multi-objective Particle Swarm Optimization (PSO) approach is proposed for summarizing web reviews, utilizing Discrete and Continuous variants to enhance scalability and performance. This method generates concise, sentiment-rich summaries and outperforms existing feature-based summarization techniques [145]. Lovinger et al. [110] developed *Gist*, a system that rapidly condenses extensive text, like product reviews, into key sentences using unsupervised learning and sentiment analysis. This adaptable framework outperforms current summarization methods across various domains.

2.2 Abstractive Text Summarization

Abstractive text summarization generates summaries by rephrasing and condensing the main ideas of the original text into new sentences, offering a more coherent and human-like summary than extractive methods [53]. This section highlights notable existing methods.

2.2.1 Advantages

Abstractive text summarization offers several benefits compared to extractive methods. It generates summaries that mimic human writing, allows for greater text condensation, and employs original wording through paraphrasing and compression [203].

2.2.2 Disadvantages

Generating high-quality abstractive summarizers is difficult due to the dependence on advanced natural language generation technology, which is still under development. These systems need to fully comprehend the input text to generate new sentences, but they often face issues with repeating words and managing out-of-vocabulary terms, leading to challenges in consistently producing effective summaries [75].

2.2.3 Major Approaches in Abstractive Summarization

Abstractive text summarization methods can be categorized into three types: structure-based approaches that use predefined elements such as graphs, trees, rules, templates, and ontologies; semantic-based methods that leverage semantic representations and natural language generation systems; and deep-learning-based techniques that apply neural network models [68]. Kavita et al. [60] introduced a technique named “Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions” for creating concise summaries. Their method involved representing each unique word as a node, with edges representing sentences, and scoring potential paths. However, this approach faced difficulties in capturing synonymous words or phrases, as they were treated as separate nodes. The advancement of sequence-to-sequence models by Sutskever et al. [185] has significantly improved the state-of-the-art in abstractive summarization. The integration of an attention-based encoder from Bahdanau et al. [16] with a beam-search decoder on the DUC 2004 dataset has demonstrated notable performance gains in abstractive text summarization, particularly. Chopra et al. [38] developed a novel architecture known as the Convolutional Attention-based Conditional Recurrent Neural Network (CARCNN) for abstractive text summarization and evaluated its performance using the Gigaword Corpus and the DUC 2004 datasets. Nallapati et al. [129] proposed an abstractive text summarization method using “Attentional Encoder-Decoder Recurrent Neural Networks”. Their model encodes the input document with a bidirectional recurrent neural network to capture contextual information and generates the summary one word at a time during decoding. See et al. [174] developed pointer-generator networks combined with a coverage mechanism to address the challenges related to out-of-vocabulary (OOV) words and repetitive phrases in summaries. This

approach allows the model to directly copy words from the source text and uses the coverage mechanism to mitigate repetition. Anh and Trang [11] further refined this model by incorporating pre-trained non contextual word embeddings, Word2Vec [125] and FastText [25], to improve the semantic representation of words. Li et al. [99] proposed a sequence-to-sequence encoder-decoder framework that includes a deep recurrent generative decoder (DRGD) to identify the latent structural elements in target summaries. They apply neural variational inference to handle complex posterior inference for latent variables. Their method combines both generative latent structures and deterministic factors in the summary generation process. Evaluation on various benchmark datasets shows that their approach outperforms current leading methods in generating abstractive summaries. A neural network model with a novel intra-attention mechanism and a combined training approach using supervised and reinforcement learning, which enhances the coherence and quality of generated summaries compared to earlier methods [140]. Song et al. [178] introduced an “LSTM-CNN based ATS framework (ATSDL)” for text summarization, which integrates Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to achieve enhanced performance. Their approach addresses the issue of rare words by focusing on fine-grained semantic phrases extracted from source sentences, resulting in improved summarization quality. Wei et al. [206] proposed a regularization method for sequence-to-sequence models to improve semantic consistency in abstractive text summarization, introducing a novel human evaluation approach and two techniques for generating a more refined output distribution, resulting in enhanced model performance and accuracy. A novel convolutional seq2seq model for abstractive text summarization is presented [225], featuring gated linear units (GLU) and residual connections to boost efficiency, a hierarchical attention mechanism for generating both keywords and key sentences, and a copying mechanism to address out-of-vocabulary terms. This model consistently outperforms current state-of-the-art approaches on the GigaWord and DUC datasets. A regularization technique for sequence-to-sequence models is introduced, which enhances semantic consistency by 4% [206]. Additionally, a new human evaluation method is proposed to more accurately assess the alignment of summaries with the source content. The method advances abstractive text summarization by integrating a contextual network for retrieving relevant content and a pre-trained language model for generation, along with a novelty metric optimized through policy learning, resulting in summaries with superior abstraction and strong performance in ROUGE scores and human evaluations [91]. Yang & Wang et al. [212] developed PGAN-ATSMT, a generative adversarial network integrated with multi-task constraints to advance abstractive text summarization, achieving superior results by producing more informative, grammatically precise, and innovative summaries compared to existing methods. A novel hierarchical human-like deep neural network

(HH-ATS) proposed for abstractive text summarization that mimics human reading and summarization techniques using a knowledge-driven attention module, multitask learning, and dual discriminator adversarial networks [211]. Rehman et al. [158] introduced an abstractive text summarization model utilizing a GRU-based encoder-decoder architecture for abstractive text summarization. The Bahdanau attention mechanism has been incorporated to address the challenge of managing long sequences in the input text.

2.2.4 Leveraging Pre-Trained Language Models and Large Language Models in Abstractive Summarization

Recently, Pre-trained Language Models (PLMs) that generate contextual embeddings have recently gained significant attraction, yielding state-of-the-art results in various NLP tasks, a success further enhanced by developments in transformer architecture [195]. These models are initially trained on large datasets and then we can fine-tune for specific applications. Radford et al. [152] introduced Generative Pre-Training (GPT), a method that combines unsupervised pre-training with supervised fine-tuning to enhance language understanding. The transformer architecture and its bidirectional encoder model **BERT** have significantly advanced performance in NLP tasks, including text summarization [50]. Bidirectional Encoder Representations from Transformers (BERT) utilizes a deep bidirectional transformer encoder to understand context from both directions by applying a masked language modeling approach. It can be fine-tuned for various tasks, such as sentence and token labeling, on new datasets. Integrating knowledge graphs (KGs) with BERT enhances the model's ability to capture lexical, syntactic, and knowledge-based information [226]. Additionally, specialized deep neural models, like Pre-training with Extracted Gap-sentences for Abstractive (PEGASUS), which is trained on large corpora with a focus on gap sentence generation, have been evaluated across various summarization tasks [222]. To address the limitation of input size in BERT-based models, the BERT windowing approach can be utilized [3]. Raffel et al. [153] developed Text-To-Text Transfer Transformer (T5), an encoder-decoder model pre-trained on a range of tasks by converting all tasks into a text-to-text format. Bidirectional Auto-Regressive Transformers (BART) [98] is a seq2seq transformer model that integrates a bidirectional BERT-based encoder with an auto-regressive GPT-style decoder. During pre-training, BART model modifies the order of original phrases randomly and employs a unique in-filling technique, where entire text spans are replaced with a single mask token. The pre-trained BART model shows exceptional performance in text generation when fine-tuned and is also highly effective in comprehension tasks. Predicting Future N-gram for Sequence-to-Sequence Pre-training (ProphetNet)

[148] used the foundational transformer-based encoder-decoder framework introduced by Vaswani et al. [195], integrating a unique self-supervised objective known as n-gram prediction. Instead of predicting the next token, it predicts the next n-tokens simultaneously, which enhances its ability to generate fluent and coherent text. Rehman et al. [159] analyzed several pre-trained models, including google/pegasus-cnn-dailymail, T5-base, and facebook/bart-large-cnn, to evaluate their performance in text summarization across various datasets.

Large Language Models (LLMs) have notably advanced abstractive summarization, receiving high praise from human evaluators [64, 146]. Liu et al. [108] introduced a new text summarization method that leverages LLMs as benchmarks, combining LLM-based evaluations with contrastive learning, enabling smaller models to perform comparably to LLMs in automated assessments, though they still lag in human evaluations. Zhang et al. [220] developed **SummIt**, an iterative summarization framework utilizing LLMs like **ChatGPT** to refine summaries through self-evaluation and feedback, improving accuracy, relevance, and control. Large Language Model Meta AI (LLaMA)-2 was utilized for biomedical text summarization on edge devices, focusing on memory optimization and reducing inference time through LoRA quantization [144].

2.2.4.1 Advantages of PLMs / LLMs

PLMs acquire contextual knowledge from extensive corpora, enhancing their comprehension of language. They are versatile, allowing them to outperform across a broad spectrum of NLP tasks. Similarly, LLMs leverage extensive training on diverse data to grasp nuanced context and semantics. It can be adapted to specific tasks with minimal additional data, enhancing their utility.

2.2.4.2 Disdvantages of PLMs LLMs

Training and fine-tuning PLMs and LLMs require substantial computational resources and energy. Storing and running large models demand significant memory and storage capacity.

2.3 Hybrid Approaches to Text Summarization

It integrates extractive and abstractive approaches through these stages: pre-processing, extracting key sentences, then generating the summary with abstractive techniques, and post-processing to validate the final summary [24, 109].

2.3.1 Advantages

It merges extractive and abstractive techniques, utilizing their combined advantages to improve summarization.

2.3.2 Disadvantages

The quality of summaries generated by a hybrid approach may be lower than those from a purely abstractive method, as they rely on extracted sentences rather than directly creating content from the original text.

2.3.3 Significant Works on Hybrid Approaches

A hybrid method is proposed for Punjabi text summarization, integrating conceptual, statistical, location-based, and linguistic features. This approach includes four novel location-based features and two new statistical measures: entropy and Z score [69]. A hybrid approach for Arabic text summarization that leverages domain knowledge and genetic algorithms, achieving enhanced outcomes by incorporating domain-specific features [4]. The approach integrates customized fuzzy features with a neural sequence-to-sequence model and attention mechanisms to improve the quality and relevance of summaries [168]. A hybrid summarization approach clusters sentences with Markov clustering, ranks them, combines top sentences using linguistic rules, and generates summaries through classification-based compression [169]. A new classification summarization framework tackles these issues by categorizing tweets and applying a two-step extractive-abstractive process to create clear and informative summaries, demonstrating greater effectiveness than current methods [167]. Alami & Mallahi, et al.[5] proposed a graph-based Arabic summarization system that combines statistical and semantic analysis with ontology structures, using *PageRank* [29] for ranking and an adapted Maximal Marginal Relevance method to address redundancy and diversity.

2.4 Approaches for Extracting Research Highlights and Summarizing Scientific Papers

This section will discuss various existing approaches for summarizing scientific papers and generating research highlights.

Scientific papers are typically longer and structurally distinct from news articles, featuring a well-defined discourse structure. They are characterized by a consistent layout, predictable placement of key components, and the use of specific cue words,

resembling a template-like format [83]. Scientific paper summarization can be broadly categorized into two types: abstract generation from the paper and summary generation based on the citation [8]. In the past, extractive summarization methods have been widely used for summarizing scientific articles.

Early work on extractive summarization of scientific documents was done with limited datasets, such as one of 188 document and summary pairs [92] where all the documents were gathered from 21 scientific/technical publications. It utilized the first trainable machine learning method. Innovative technique proposed by Paice et al. [138] for automatic abstract generation by identifying phrases within academic texts that inherently highlight the significance of the content, thus enabling the extraction of key information for summary creation. A summarizing technique that focuses on the rhetorical status of assertions in 80 scientific articles, part of a larger corpus of 260 articles, has been developed by Teufel and Moens [188]. A sentence-based automatic summarizing system has been built based on feature extraction and query-focused methods [199]. Mohammad et al. [127] explored how abstracts and citation texts contribute to generating technical surveys. They found that citation texts offer valuable, often missing insights not captured by abstracts, improving the comprehensiveness of multi-document summaries. This model used a set of features to rank sentences for scientific paper summarization. Contractor et al. [45] proposed a model for extractive summarization to utilize the concept of argumentative zones (AZs) framework for academic papers. Lloret et al. [109] developed the COMPENDIUM system for generating biomedical research abstracts, utilizing both extractive and hybrid extractive-abstractive models to evaluate their effectiveness and suitability for summarization. To better deal with the long text of a research paper in abstractive summarization, a multiple timescale model of the gated recurrent unit (MTGRU) has been used in [85]. They have contributed a new corpus containing pairs of (introduction, abstract) of computer science papers from [arXiv.org](https://arxiv.org). Souza et al. [179] have proposed a multi-view extractive text summarization approach for long scientific texts. Recent advancements have attempted to summarize entire research papers, focusing specifically on the generation of the paper title from the abstract (title-gen) and the generation of the abstract from the body of the paper (abstract-gen) in biomedical domain [136]. A framework for summarizing scientific papers that enhances citation accuracy and integrates contextual information, showing significant improvements over existing methods in biomedical and computational linguistics [42]. Cohan et al. [41] proposed abstractive model for generating abstract from scientific papers and proposed arXiv and PubMed datasets. A method for automatic event-based summarization using a 5W1H framework is proposed [221]. This approach ranks top-k sentences by relevance and importance, producing detailed, event-focused

summaries that enhance the accessibility and utility of scientific and technical information. Kinugawa and Tsuruoka [86] proposed a two-level hierarchical structure based on encoder-decoder for extractive summarization of research papers.

Generating research highlights from scientific articles is different than document summarization. Collins, Augenstein and Riedel [43] have developed supervised machine learning methods to identify relevant highlights from the full text of a paper using a binary classifier. They also contributed a new benchmark dataset of URLs, which includes $\sim 10K$ articles from computer science domain, labelled with relevant author-written highlights. L. Cagliero et al. [33] proposed an extractive approach based on gradient boosting method to select the top- k most relevant sentences as a research highlights, unlike a simple binary classification of sentences as highlights or not. Note that this is also extractive in nature.

2.5 Reducing Hallucinations and Ensuring Factual Accuracy in Text Summarization

In this section, we intend to explore various methods and strategies that have been developed to reduce hallucinations and enhance factual consistency in abstractive summarization models.

Early summarization models used extractive approaches, which struggled with rephrasing and combining information. Abstractive techniques, advanced by Recurrent Neural Networks (RNNs), improved the process but still face issues like inaccuracies, out-of-vocabulary (OOV) words, and repetition [129]. The pointer-generator model with a coverage mechanism addresses issues with OOV words and helps to reduce repetitive word generation [174]. Large pre-trained transformer models perform well on natural language tasks [50, 160], but they struggle with long textual sequences. These documents often exceed the maximum context length of standard transformers, requiring specialized architectural changes and training methods. These systems often have difficulty ensuring that generated summaries are strictly based on the source document, avoiding the inclusion of fabricated or hallucinated information. Such hallucinations may arise from factors such as biases in the training data, insufficient contextual understanding, or excessive model optimization. Cao et al. [34] and Kryściński et al. [90] found that about 30% of summaries generated by seq2seq models exhibit hallucinations. Consequently, the NLP community has increasingly focused on improving the faithfulness and factual accuracy of abstractive summarization [63, 90, 230]. Recent research has investigated hallucination issues at the entity and relation levels in generated text [90]. Nan et al. [130] proposed a method to reduce entity hallucination

by employing a data filtering technique during training and utilizing multi-task learning. Goyal and Durrett [63] explored relation hallucination, focusing on whether the semantic relationships indicated by dependency arcs in a generated sentence align with those in the source sentence. Narayan et al. [131] use entity chain content planning to enhance summary faithfulness. There is increasing interest in quantitatively assessing text generation models' faithfulness. Current metrics like ROUGE [102], METEOR [17], BERTscore [224], and MoverScore [227] often do not align well with human judgments of faithfulness [90]. Recent research has focused on categorical and content-based methods for evaluating summary faithfulness [63].

2.6 Datasets for Text Summarization

This section highlights the most commonly used and popular benchmarking datasets for ATS systems.

2.6.1 News Article Datasets

1. **Document Understanding Conference (DUC) dataset:** Provided by NIST, these datasets are crucial for text summarization research, covering DUC 2001 to DUC 2007 and focusing on English news domain data. DUC 2004 available in Arabic language also. They include manually created summaries, baseline summaries, and those from challenge participants. Access requires submitting application forms via the DUC website ¹. Although valuable for evaluating ATS systems, they do not provide enough data for training neural network models [104].
2. **Text Analysis Conference (TAC) dataset:** In 2008, the summarization track from DUC was incorporated into TAC and focusing on English news domain data. Accessing the TAC datasets necessitates completing application forms available on the TAC website ².
3. **Gigaword 5 dataset:** This dataset, containing approximately ten million English news documents, is commonly used for abstractive summarization and is suitable for training and testing neural network models. A limitation is that it provides only headlines as summaries [129].
4. **CNN/Daily Mail dataset:** Initially designed for passage based question answering [74], this corpus is now extensively used for evaluating ATS systems and

¹<https://www-nlpir.nist.gov/projects/duc/data.html>

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

focusing on English news domain data. It is a modified version includes multi-sentence summaries for evaluating abstractive summarization [129].

5. **NEWSROOM (CORNELL NEWSROOM)** : This dataset, introduced by Grusky et al. [67], comprises a vast collection of articles and their summaries, contributed by authors and editors from 38 major news publications.
6. **New York Times Annotated Corpus** : Introduced by Sandhaus et al. [172], contains over 650,000 articles from the New York Times, accompanied by summaries written by library scientists.
7. **XSum dataset** : Introduced by Narayan et al. [131] consists of more than 226K wide variety of news articles collected from **BBC** along with one-sentence summary.
8. **Multi-News dataset** : Introduced by Fabbri et al. [55], includes news articles and professionally written summaries by editors.

2.6.2 Datasets for Scientific and Research Articles

1. **CSPubSum dataset** : Introduced by Collins et al. [43], includes URLs for 10,147 computer science papers from ScienceDirect³, with highlights serving as the golden summaries along with the full text of each paper.
2. **PeerRead dataset** : Introduced by Kang et al. [81], it comprises paper drafts submitted to top conferences such as NIPS, ICLR, and ACL including their acceptance or rejection decisions.
3. **PubMed dataset** : Introduced by Cohan et al. [41], it comprises biomedical and life sciences papers, with summaries provided as abstracts.
4. **ScisummNet dataset** : Introduced by Yasunaga et al. [214], it comprises around 1,000 scientific papers with manual annotations for summarization.
5. **TalkSumm dataset**: Introduced by Lev et al. [97], features automatically generated summaries of scientific papers presented at international conferences such as EMNLP, NAACL, and ACL. The dataset is continuously expanded using video transcripts of conference talks.
6. **SCITLDR dataset** : Introduced by Cachola et al. [32], contains 5,411 TLDRs for computer science papers, derived from authors' TLDRs on OpenReview2 and

³<https://www.sciencedirect.com/>

peer review comments rewritten by experts. The dataset features multiple gold summaries per paper to address variations in human-written summaries.

2.6.3 Legal Article Datasets

1. **BigPatent dataset** : Introduced by Sharma et al. [175], this dataset comprises a vast collection of U.S. patent documents along with their corresponding human-authored abstractive summaries.
2. **RulingBR dataset** : Introduced by Feijó, Pereira, and Moreira [47], this dataset comprises around 10,000 rulings from the Brazilian Federal Supreme Court.
3. **BillSum dataset**: Introduced by Kornilova and Eidelman [89], this dataset contains 22,218 U.S. Congressional bills, with 1,237 specifically from California.
4. **AustLII dataset** : Introduced by Greenleaf et al. [65], this resource offers online access to Australasian legal materials, featuring more than 1,500 Commonwealth statutes and regulations, in addition to various law reform reports.

2.6.4 Social Media Datasets

1. **Webis-TLDR-17 dataset**: Introduced by Volske et al. [200], this dataset comprises articles from Reddit, a social media platform, along with the summaries authored by the original writers.
2. **Reddit TIFU dataset** : Introduced by Kim et al. [84], this dataset includes 120,000 posts from the TIFU subreddit, collected between January 2013 and March 2018, offering a substantial collection of informal, crowd-generated content from a diverse online discussion platform.
3. **TWEETSUM dataset**: Introduced by He et al. [73], is a large-scale dataset with 44,034 tweets and 11,240 users from 12 events, featuring social signals, user relationships, and expert summaries
4. **LCSTS dataset** : A dataset from ‘Sina Weibo microblogging site’, contains over 2 million Chinese short texts with summaries, and features 10,666 manually annotated text-summary pairs [76].

2.7 Evaluation Metrics for Text Summarization

This section discusses the methods employed in evaluating text summarization models, with a primary focus on automated techniques for summary assessment.

Basically, there are two evaluation methods for measuring generated summaries [70]:

- **Intrinsic methods** assess summary quality through human evaluation, focusing on coherence and informativeness.
- **Extrinsic methods** evaluate summary quality based on task-specific performance, such as information retrieval, to determine the summaries' utility in practical applications like relevance assessment and reading comprehension. Text summarization evaluation includes manual and automatic methods.

Since manual evaluation is costly, automatic methods are needed to assess the performance of automatic text summarization systems.

Common metrics for automatic text summarization evaluation are ROUGE [102], METEOR [17], MoverScore [227], and BERTScore [224].

2.7.1 ROUGE

When comparing the model-generated research highlights (`ModelHighlights`) with the author-written research highlights (`AuthorHighlights`) for evaluation, ROUGE- n calculates the recall, precision, and F1-measure for each model using Equations (2.1), (2.2) and (2.3). Note that an n -gram is a contiguous sequence of n words from a piece of text.

Recall (R) is defined as:

$$R = \frac{\#matched\ n - grams\ in\ (\text{ModelHighlights}, \text{AuthorHighlights})}{\#n - grams\ in\ \text{AuthorHighlights}} \quad (2.1)$$

Precision (P) is defined as:

$$P = \frac{\#matched\ n - grams\ in\ (\text{ModelHighlights}, \text{AuthorHighlights})}{\#n/grams\ in\ \text{ModelHighlights}} \quad (2.2)$$

F1-measure ($F1$) is calculated using the formula:

$$F1 = 2 * \frac{R * P}{R + P} \quad (2.3)$$

We have used ROUGE-1, ROUGE-2 and ROUGH-L. In particular, ROUGE-L measures the longest matching sub-sequence of words between the two strings. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script [102].

2.7.2 METEOR

METEOR assigns a score to the match based on a combination of unigram precision, unigram recall, and a fragmentation measure that is intended to directly represent how well-ordered the matched words in the model-generated research highlights and author-written research highlights are. It calculates recall (R) and precision (P) of unigrams based on Equations (2.1) and (2.2), respectively. Next, it computes F_{mean} score and chunk penalty using the formula (2.4) and (2.5):

$$F_{mean} = \frac{10(R * P)}{R + 9P} \quad (2.4)$$

$$Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams_matched} \right)^3 \quad (2.5)$$

where *chunks* is defined as a set of unigrams that are adjacent in the **ModelHighlights** and in the **AuthorHighlights**. The final METEOR score is computed as follows:

$$Score = F_{mean} * (1 - Penalty) \quad (2.6)$$

2.7.3 MoverScore

MoverScore [227] is calculated based on the contextualized representations and Word Mover’s Distance (WMD) [93] between the research highlights generated by model (**ModelHighlights**) and the research highlights written by authors (**AuthorHighlights**). It can take into account the presence of new or unseen words in the generated text, and evaluate how well they fit into the overall structure and content of the original text. It allows many-to-one alignment to map the semantically similar words in **ModelHighlights** and **AuthorHighlights**, whereas BERTScore considers only one-to-one alignment. The sentences of the author-written research highlights (**AuthorHighlights**) and model-generated research highlights (**ModelHighlights**) are represented as x and \hat{x} . Their sequence of n -grams are denoted as x^n and \hat{x}^n . The transportation cost matrix (C) is calculated based on a distance metric (d) between the n -grams as follows:

$$C_{i,j} = d(x_i^n, \hat{x}_j^n) \quad (2.7)$$

where $d(x_i^n, \hat{x}_j^n)$ is the Euclidean distance between the i -th n -gram of x and the j -th n -gram of \hat{x} where both the n -grams are represented by their respective embeddings. The authors in [227] define a transportation flow matrix F where $F(i, j)$ captures the amount of flow from the i -th n -gram (x_i^n) in x^n to the j -th n -gram (\hat{x}_j^n) in \hat{x}^n . Let $\langle C, F \rangle$

denote the sum of all elements in the matrix obtained from element-wise multiplication of C and F . We associate weights f_{x^n} and $f_{\hat{x}^n}$ with the n -grams x_n and \hat{x}_n , such that each n -gram gets a single weight value in each case and assume that each of f_{x^n} and $f_{\hat{x}^n}$ defines a probability distribution (i.e., the entries of each vector sums to 1). Finally, the moverscore [227] is defined as

$$\text{WMD}(x^n, \hat{x}^n) = \min_{F \in \mathbb{R}^{|x^n| \times |\hat{x}^n|}} \langle C, F \rangle \quad \text{such that } F \mathbf{1} = f_{x^n} \text{ and } F^\top \mathbf{1} = f_{\hat{x}^n} \quad (2.8)$$

2.7.4 BERTScore

For BERTScore computation, we consider the cosine similarity of contextual embeddings of each word from model-generated research highlights and author-written research highlights, instead of counting the exact words matched across them. Denoting the contextual embeddings of the author-written research highlights by $\vec{x} = \langle \vec{x}_1, \dots, \vec{x}_n \rangle$ and those of the model-generated research highlights by $\hat{\vec{x}} = \langle \hat{\vec{x}}_1, \dots, \hat{\vec{x}}_m \rangle$, the recall (R_{BERT}), precision (P_{BERT}), and F1-scores ($F1_{\text{BERT}}$) are calculated as follows:

$$R_{\text{BERT}} = \frac{1}{m} \sum_{\vec{x}_i \in \vec{x}} \max_{\hat{\vec{x}}_j \in \hat{\vec{x}}} \vec{x}_i^\top \hat{\vec{x}}_j \quad (2.9)$$

$$P_{\text{BERT}} = \frac{1}{n} \sum_{\hat{\vec{x}}_j \in \hat{\vec{x}}} \max_{\vec{x}_i \in \vec{x}} \vec{x}_i^\top \hat{\vec{x}}_j \quad (2.10)$$

$$F1_{\text{BERT}} = 2 * \frac{R_{\text{BERT}} * P_{\text{BERT}}}{R_{\text{BERT}} + P_{\text{BERT}}} \quad (2.11)$$

These metrics compare generated summaries with reference summaries to assess quality. However, these metrics are insufficient for measuring factual consistency [90]. Therefore, we also employed three new metrics introduced by [130] to evaluate the factual accuracy of the generated summaries.

2.8 Summary

Reviewing the literature reveals both advancements and limitations. For **RQ1**, **RQ2**, and **RQ3**, it is clear that while abstractive summarization systems for scientific research papers are crucial for producing summaries, or research highlights or key findings, they present considerable challenges. Therefore, our focus will be on addressing these issues.

For **RQ4**, it is observed that fine-tuning pre-trained language models can result in hallucinations. Although some research has tackled hallucination reduction in news datasets, strategies for addressing this issue in scientific document summarization or

research highlights generation are still not well-defined. Thus, our focus will be on addressing this issue.

For **RQ5**, creating extensive datasets with human-written summaries is both time-consuming and costly. Thus, there is an urgent need for a new automated dataset to support data-hungry deep learning models. Creating extensive datasets with human-written summaries is both time-consuming and costly. Thus, there is an urgent need for a new automated creation of dataset to support data-hungry deep learning models. Assessing abstractive summarization models is both essential and complex. It requires evaluating how effectively the models generate summaries that are coherent, informative, and uniquely phrased, while also correlating with human judgments.

3

Generation of Research Highlights with Deep Learning: Exploring the Impact of Embedding Types on Model's Performance

This chapter explores the generation of research highlights using deep learning techniques, with a particular focus on the impact of different embedding types. The study is based on our three contributed fundamental papers, each contributing to the advancement of abstractive approaches for extracting research highlights from scientific papers.

3.1 Introduction

Scientific publications are growing at an exponential rate [27]. It has been reported that the number of scientific articles doubles roughly every nine years [194]. This rapid growth presents a significant challenges for researchers who struggle to stay up-to-date about the latest advancements. Even in a limited sub-field, researchers find it very challenging to keep track of the cutting edge of research. Therefore, to make it easier for researchers to appreciate the main key findings of a paper, publishers have adopted many novel presentation techniques. To addresses this issue, one recent trend is to complement the abstract of a paper with *research highlights* along with the full text.

Research highlights – a bulleted list of points summarizing the main findings of the paper. Research highlights are typically written by the author along with the abstract. They are often easier to read and grasp than a longer paragraph, especially on hand-held devices. Moreover, research highlights can be used by search engines for indexing the articles and subsequently, retrieve or recommend them to the intended users. The main findings and contributions of the paper can be emphasized in promotional materials such as social media posts by utilizing the highlights. Yet, not all scholarly articles contain research highlights written by the authors. Research highlights and abstract are both *summaries* of the research paper.

3.2 Background and Motivation

This section provides an overview of the background related to research highlight generation and the motivation behind this research.

3.2.1 Background

Text summarization is a process to present the gist of a source document or a set of related documents. The main benefit of text summarization is that it reduces the amount of time the reader has to spend to extract the main information in the document. Extractive summarization and abstractive summarization are two broad approaches used in automatic text summarization [112, 128]. Extractive approaches [92] simply copy some relevant sentences from the documents and ignore the rest. Abstractive approaches [53] can induce new relevant words in the summary in the same way that a person does – they first read the entire text, comprehend it, and then summarize using suitable new words. Therefore, abstractive approaches typically provide better summaries compared to those produced by extractive methods.

3.2.2 Motivation

The huge publication of scientific literature has made it increasingly difficult for researchers to stay updated with the latest developments. Traditional abstracts, while providing a summary of research papers, can be lengthy and may not always communicate the core findings clearly or effectively. Research highlights address this limitation by offering a concise summary of the main contributions in a bullet-point format, which is more easily readable on handheld devices.

However, generating high-quality research highlights presents several challenges. Extractive methods may produce summaries that are disjointed or lack of context, while

abstractive methods require sophisticated models capable of understanding and rephrasing complex content. Recent advancements in deep learning provide new opportunities to enhance summarization by exploring various embedding types that can improve the relevance and coherence of generated highlights.

The main goal of this chapter is to tackle the challenges associated with generating research highlights by utilizing deep learning techniques. It specifically highlights how different types of embeddings impact the effectiveness of abstractive summarization models performance. By doing so, the chapter aims to advance the field of summarization and improve the generation of research highlights, helping researchers keep up with the growing volume of scientific literature.

3.3 Challenges and Opportunities

This section discusses the current challenges in generating research highlights and explores potential advancements in the field through innovative methods and technologies.

3.3.1 Challenges

Scientific papers often present significant challenges due to their length and complexity, making it difficult to create concise and relevant highlights. Abstractive summarization models need to manage extensive text efficiently while preserving accuracy and coherence. Generating highlights that accurately capture the main contributions of a paper is a demanding task. The generated highlights must be both informative and directly aligned with the core findings of the research.

3.3.2 Opportunities

Recent progress in deep learning and various types of embeddings, provides new opportunities to improve summarization quality. Assessing and comparing various embedding techniques can provide insights into their effects on summarization performance. Improvements in embeddings can help models better understand and generate highlights. Additionally, combining summarization models with new technologies, like search engines and recommendation systems, can make research highlights more accessible and visible.

3.4 Different Text Representations Methods

Word embedding is a method of representing each word as a vector of floating-point numbers. Word embeddings map words into n-dimensional vector spaces where words

with similar meanings (such as ‘doctor’ and ‘physician’) or words related by context (like ‘doctor’ and ‘hospital’) are positioned closer together, based on the training data. Different techniques are used depending on the specific requirements of the model and dataset [204].

3.4.1 Traditional Approaches for Text Representation

In this subsection, we have discussed traditional approaches to text representation. NLP can be divided into two key areas: Natural Language Understanding (NLU) and Natural Language Generation (NLG). NLU focuses on interpreting the meaning of text, while NLG involves generating text for communication, either to express ideas or respond. NLG is challenging due to the need for precise word choice, flexible word order, and ensuring fluency and grammatical correctness.

3.4.1.1 One-Hot Encoding

One-hot encoding is a straightforward approach in NLP for representing words as vectors. In this method, each word in the vocabulary is assigned a unique vector, where the vector’s length matches the total number of words in the vocabulary. All elements in the vector are zeros, except for a single element corresponding to the word’s position in the vocabulary, which is set to one. This method ensures that each word has a distinct representation, with only one active element in its vector, giving it the name “one-hot” encoding [205].

One-hot encoding generates large, sparse vectors, which can be costly in terms of computation and memory, especially with extensive vocabularies. It does not reflect the semantic relationships between words and is confined to the vocabulary used in training, making it ineffective for handling new or unknown words.

3.4.1.2 Bag-of-Words (BoW)

The Bag-of-Words (BoW) model encodes a sequence by averaging the vectors of its words, effectively using their average position in the vector space. This approach assigns equal weight to all words, potentially neglecting the significance of certain words in understanding the meaning. BoW represents a document as a collection of words and their frequencies, without considering the order of the words, resulting in a vector based purely on word counts [71]. The BoW model has the advantage of being simple and interpretable. However, it has notable disadvantages, ignores the sequence of words, which can impair performance on tasks that require sequential information, and it generates sparse representations with many zero values, leading to increased memory usage and computational inefficiency, especially with large datasets.

3.4.1.3 Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF refines the Bag-of-Words model by adjusting word frequencies with the Inverse Document Frequency (IDF). IDF calculates the importance of a term using the logarithm of the ratio of the total number of documents to the number of documents containing the term. This method highlights unique terms and reduces the influence of common ones [80]. TF-IDF is useful for information retrieval but has drawbacks, such as neglecting semantic relationships between words, favoring longer documents with higher term frequencies, and overlooking important terms.

3.4.2 Neural Approaches for Text Representation

Models that use categorical word representations struggle to capture the syntactic and semantic meanings of words and also face challenges due to high dimensionality. These drawbacks led researchers to investigate distributed word representations within lower-dimensional spaces [26]. Such vector-based representations enable learning algorithms to perform better in natural language processing tasks by grouping similar words together. Neural networks are used to learn these distributed representations. Word2Vec employs two neural embedding techniques: Continuous Bag of Words and Skip-Gram [204]. Continuous Bag of Words (CBOW) predicts a target word based on the surrounding context within a specified window, using the sequence of words around it. In contrast, Skip-Gram aims to predict context words from a given target word. It uses the target word as input to a log-linear classifier with a continuous projection layer, predicting words within a certain range around the target. While CBOW focuses on context-to-target prediction, Skip-Gram emphasizes target-to-context prediction, resulting in more meaningful embeddings. Skip-gram performs well with smaller training datasets and effectively represents rare words. In contrast, CBOW is more efficient and handles frequent words better.

3.4.2.1 Pre-trained Word-Embeddings

The CBOW and Skip-gram models significantly advanced distributed word representations. Building on these, GloVe and FastText further improved word embeddings. While Word2Vec captures word semantics based on local context, it lacks global statistical insights. GloVe enhances this by using global co-occurrence matrices to better capture word relationships, making it highly effective for text classification [132].

3.4.2.2 Global Vectors for Word Representation(GloVe)

This section explores how unsupervised methods use word occurrence statistics to generate meaning in word vectors and introduces GloVe [142], a model that directly captures global corpus statistics for word representations. Formally, let X represent the word co-occurrence count matrix, with X_{ij} denoting the frequency of word w_j appearing in the context of word w_i . The loss function is then defined as follows:

$$L = \sum_{i,j=1}^{|V|} g(X_{ij}) \left(C(w_i)^\top C(w_j) + b_i + b_j - \log X_{ij} \right)^2 \quad (3.1)$$

where $|V|$ is the vocabulary size, $g(X_{ij})$ is a weighting function, $C(w_i)$ and $C(w_j)$ are the word vectors for w_i and w_j , and b_i and b_j are bias terms for the words w_i and w_j .

This method does not handle out-of-vocabulary (OOV) words, either missing from the vocabulary or training corpus, prompting the development of models to address this issue, one of which we briefly describe in section 3.6.

3.4.2.3 FastText

The FastText [25] approach enhances the skip-gram model by incorporating character n -grams for word representation. Words are represented as the sum of vectors for their n -grams, enabling efficient training on large corpora and producing embeddings for out-of-vocabulary words. Evaluation across nine languages on word similarity and analogy tasks demonstrates that this method outperforms recent morphological word representations.

While pre-trained word embeddings like **GloVe** and **FastText** have achieved satisfactory results, they fall short in certain aspects. Specifically, they do not consider the sequence of words, which leads to a diminished ability to capture the syntactic and semantic nuances of sentences.

3.4.2.4 Embeddings from Language Models(ELMo)

Sentence embeddings function similarly to word embeddings, but they represent entire sentences as vectors rather than individual words. Notable state-of-the-art models for sentence embeddings include ELMo, InferSent, and SBERT.

ELMo generates word vectors based on a deep bidirectional language model (biLM) that has been pre-trained on a vast text corpus [143]. Unlike earlier techniques, ELMo integrates information from all layers of the biLM, employing a weighted combination of these vectors for each specific task. This method significantly enhances performance

compared to relying solely on the top layer of the LSTM. The word embeddings produced by ELMo are derived from the representations learned through the Bi-directional Language Model (BiLM). It generates final word vectors from a bidirectional language model that includes both forward and backward contexts. Here, (t_1, t_2, \dots, t_N) . represented as sequence of N tokens.

$$BiLM = \sum_{k=1}^N (\log p(t_k | t_1, \dots, t_{k-1}; \Theta_x, \vec{\Theta}_{LSTM}, \Theta_s) + \log p(t_k | t_{k+1}, \dots, t_N; \Theta_x, \overleftarrow{\Theta}_{LSTM}, \Theta_s)) \quad (3.2)$$

The forward and backward directions share the parameters for token representation Θ_x and softmax Θ_s . The parameters for forward and backward LSTM are denoted as $\vec{\Theta}_{LSTM}$ and $\overleftarrow{\Theta}_{LSTM}$, respectively.

ELMo extracts representations from an intermediate layer of the BiLM and performs a linear combination for each token. The BiLM contains $2L + 1$ sets of representations, given by:

$$R_k = \left\{ X_k^{LM}, \vec{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM} \mid j = 1, \dots, L \right\} = \left\{ h_{k,j}^{LM} \mid j = 0, \dots, L \right\} \quad (3.3)$$

For each BiLSTM layer, $h_{k,j}^{LM} = [\vec{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM}]$, where $h_{k,0}^{LM} = x_k^{LM}$ represents the token layer.

ELMo flattens all layers in M into a single vector, as described by the equation 3.4

$$ELMo_{task}^k = \mathbb{E}(R^k; \Theta_{task}) = \gamma^{task} \sum_{j=0}^L s_j^{task} h_{k,j}^{LM} \quad (3.4)$$

To merge representations from various layers, the weights s^{task} are normalized using softmax. Meanwhile, γ^{task} serves as a hyperparameter for scaling and optimization these representations.

3.4.2.5 Transformer-based Pre-trained Language Models

Generative Pre-Training Transformer(GPT) uses the transformer architecture [195] to understand and generate text [152]. It learns from a vast amount of text data, allowing it to handle language tasks effectively. Unlike ELMo, which uses bidirectional context, GPT is an auto-regressive model that predicts the next word based on the words that came before it. While GPT performs well on many tasks, it only processes text from left to right, which can be a limitation.

BERT (Bidirectional Encoder Representations from Transformers) enhances GPT

by incorporating bidirectional context and employing two tasks—masked language modeling and next sentence prediction—for improved text understanding and prediction [50]. OpenAI’s GPT-2, released in 2019, is an upgraded GPT with improved layer normalization and residual connections across various sizes. XLNet, or generalized auto-regressive pre-training for language understanding, leverages a permutation-based approach to predict bidirectional context, addressing limitations of BERT’s masked language model [213]. RoBERTa [107] refines BERT by extending training with larger datasets, removing next sentence prediction, and using dynamic masking to boost performance. ALBERT [94] improves BERT’s scalability by reducing parameters through factorized embeddings and cross-layer sharing, replacing next sentence prediction with sentence order prediction to enhance efficiency. Numerous other pre-trained models for word embeddings are also available, each with distinct features and advancements.

3.5 Main Contributions

The main contributions of this chapter are:

1. We propose a mechanism to combine GloVe, ELMo, and SciBERT word embeddings with pointer-generator networks that have a coverage mechanism to automatically generate research highlights.
2. We also analyze the impact of different word embeddings on model performance for research highlights generation.
3. We analyze the performance of generating research highlights by combining different sections of research papers as input types: the input can be either the abstract alone or a combination of the introduction, conclusion, and abstract in various ways.
4. We evaluate the performance of each models using ROUGE [102], METEOR [17], MoverScore [227], and BERTScore [224] metrics.

3.6 Pointer-Generator Model with GloVe Word Embeddings and Coverage Mechanism

In this work [163], to generate research highlights from scientific papers, we have utilized an abstractive summarization model based on a pointer-generator network extended with a coverage mechanism [174] and GloVe embeddings [141]. We used pre-trained GloVe embeddings [141] to represent the tokens of the input document.

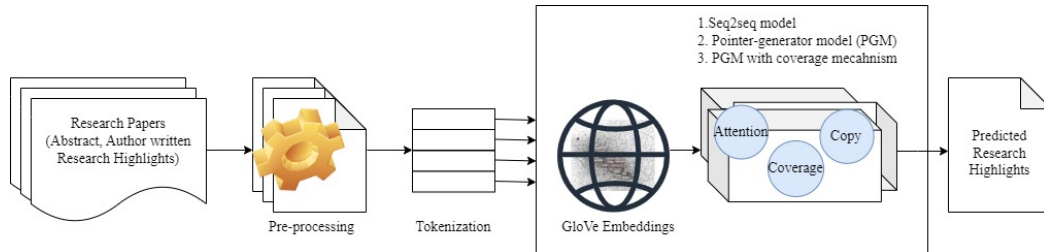


FIGURE 3.1: GloVe based encoder-decoder model for research highlight generation.

This section is organized into four major sub-sections: (i) methodology, (ii) experimental setup, (iii) evaluation results, and (iv) analysis with a case study.

3.6.1 Methodology

In this section, we have described the model we use to generate research highlights from the scientific papers.

We proposed an abstractive summarization model for generating research highlights from abstract of a paper, based on a pointer-generator network enhanced with a coverage mechanism [174] with combining GloVe embeddings [141]. Figure 3.1 illustrates the workflow of our system. We have used three deep learning-based models to generate research highlights.

Seq2seq is the *sequence-to-sequence (seq2seq) model with attention* [129]. Each abstract is tokenized and the tokens are converted to GloVe vectors [141] that are sequentially fed into the encoder which is a single-layer bidirectional Long Short-Term Memory (BiLSTM). The decoder is a single-layer unidirectional LSTM. The model uses neural attention [16] to attend to the words in the source document while generating the target words for the summary.

PGM is a *pointer-generator network* [174], which augments the above seq2seq model with a special copying mechanism. When generating words, the decoder probabilistically decides between generating new words from the vocabulary (i.e. from the training corpus) and copying words from the input abstract (by sampling from the attention distribution). While the generator helps in novel paraphrasing, copying helps to tackle out-of-vocabulary (OOV) words.

Thirdly, the second model pointer-generator augmented with coverage mechanism of Tu et al. [191] to avoid erroneously repeating the same words/phrases during decode.

3.6.2 Experimental Setup

In this subsection, we discussed the datasets used, the data pre-processing steps, and the implementation details.

3.6.2.1 Dataset Details

Our work used the CSPubSum dataset, introduced by Collins et al. [43], which includes URLs $\sim 10K$ in the field of computer science sourced from ScienceDirect ¹. We have crawled the dataset. Every document contains the following fields: title, abstract, research highlights written by the authors, a list of keywords mentioned by the authors, and various sections such as introduction, related work, experiment, and conclusion, as typically found in a research paper. Each example in the dataset is organized as (abstract, author-written research highlights). The dataset was divided into training, validation, and test subsets, following an 80 : 10 : 10 ratio. In this dataset, the average abstract size is 186 words while that of highlights is 52; for 98% of the papers, highlights are 1.5 times or more shorter than the corresponding abstract. Therefore, research highlights can be viewed as a summary of both the abstract and the full paper.

3.6.2.2 Data Pre-Processing

We have used the Stanford CoreNLP Tokenizer ² for tokenizing the sentences. The whole corpus was initially converted to lowercase. We eliminated unnecessary symbols, letters, HTML tags, parentheses, special characters, extra spaces, URLs, and other irrelevant elements. We organized it as (*abstract, research highlights written by author*). For the abstract is used as the input, we set the maximum number of input tokens to 400 and the maximum token count of the generated research highlights tokens is set to 100. The above figures are motivated by the observation that the average length of an abstract is 186, the average length of the author-written highlights in a paper is 52.

3.6.2.3 Implementation Details

We trained three variants of the proposed model: sequence-to-sequence (seq2seq) model with attention and GloVe embeddings for the input tokens (**Seq2Seq** + **GloVe**), pointer-generator network with GloVe embeddings for the input tokens (**PGM** + **GloVe**), and pointer-generator network with coverage mechanism and GloVe embeddings for the input tokens (**PGM** + **GloVe** + **Coverage**). We trained all models on Tesla T4 Colab that supports GPU-based training. We used mini-batches of size 16. For all models, we used bidirectional LSTMs with cell size of 256. For all the models, we used the same vocabulary of around 50K tokens, beam search in the decoder with size 4.

¹<https://www.sciencedirect.com/>

²<https://stanfordnlp.github.io/CoreNLP/>

3.6.3 Results

To evaluate the performance of the models, we use ROUGH [102], METEOR [17], MoverScore [227], and BERTScore [224] metrics. These are the standard metrics used to measure the performance of summarization models [56]. Table 3.1 shows that pointer-generator model with coverage mechanism with combining GloVe embeddings (**PGM + GloVe + Coverage**) always achieved highest F1-score in terms of ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore and BERTScore values.

TABLE 3.1: Evaluation of pointer-generator type models with GloVe embeddings: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore are based on abstracts as input from the CSPubSum dataset. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. All scores are presented as percentages (%).

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore
abstract only	Seq2Seq + GloVe	19.90	1.93	18.58	7.39	9.24	83.94
	PGM + GloVe	30.9	7.55	28.62	11.04	13.62	84.99
	PGM + GloVe + Coverage	31.46	8.57	29.14	12.01	15.54	85.31

3.6.4 Case Studies

In this subsection, we have shown a few research highlights generated by our models. In all case studies, yellow color represents **factual errors**, orange color represents **repeating phrases**, green color identifies **correctly added words or phrases**, and cyan color identifies **novel words**.

Figure 3.2 shows that the *Seq2Seq* model with *GloVe embeddings* generated many out-of-vocabulary (OOV) words and factual errors, such as “non-intrusive numerical tool”, “solar artery supply planning”, “cohort and accuracy...”, and many more. Pointer-generator models with GloVe embeddings generate more meaningful phrases, such as “This paper reports a multiscale asymptotic analysis” and “regularity hypothesis”. However, this model also generate factual errors, such as “expansions” and “bone microstructure samples”, though less frequently than the (*Seq2Seq + GloVe*) model. This method also repeat some phrases, such as “proposed method is robust”. It generate a novel words such as “proposed approach”. But, the *pointer generator* model with *Glove embeddings* and *coverage mechanism* reduces them. The first sentence from the *pointer generator model* with *Glove embeddings* and *coverage mechanism* contains words “without object ... properties” that do not fit into the context, but its other highlights are meaningful.

Author-written highlights:	“A novel multiscale analysis and computation is proposed. Heat transfer problem of periodic porous materials with radiation boundary condition are considered. Error estimates of the multiscale approximate solution are derived on some regularity hypothesis. Some numerical results are given in details to validate the multiscale method.”
Seq2seq+Glove:	“A non-intrusive numerical tool is developed for solar artery supply planning . The results were analyzed, based on the fe modeling of finite element model . The approaches provides practical advantages of the cohort and accuracy of concrete equipment . Agent-based fe technology neural network procedures are used for assessment assessment. Results obtained from a real composite sample are considered and discussed.”
PGM+Glove:	“This paper reports a multiscale asymptotic analysis and developed protocol. The proposed approach is based on regularity hypothesis expansions . The proposed method is robust and can achieve predicting heat transfer performance . The proposed method is robust and efficient for given bone microstructure samples .”
PGM+Glove+Coverage:	“Reports a multiscale asymptotic analysis without object propagation using minimal porous properties . Predicting heat transfer performance of periodic porous materials with radiation boundary condition. Finite element algorithms and computation of approximate convergence results .”

FIGURE 3.2: Original abstract, author-written research highlights and research highlights generated by pointer-generator type models with GloVe embeddings. The meaning of the colors (e.g., green = correct) is explained in main text. Abstract taken from <https://www.sciencedirect.com/science/article/abs/pii/S0168874X15000621>.

3.6.5 Summary of Findings

The contribution of this work, highlights the automatic generation of research highlights from abstracts of a paper using advanced deep neural network models, specifically a pointer-generator network with a coverage mechanism and GloVe embeddings. Through various experiment, the proposed abstractive approach demonstrates high performance on a large corpus of computer science papers as CSPubSum, achieving notable ROUGE score, METEOR score, MoverScore and BERTScore values. However, there remains a need for improvement, particularly in refining the generated highlights to eliminate irrelevant words or phrases. The research aims to enhance the discoverability and readability of scientific literature, ultimately facilitating more efficient knowledge dissemination and aiding researchers in navigating the large amount of published work.

3.7 Pointer-Generator Model with ELMo Word Embeddings and Coverage Mechanism

In this work [162], we further analyzed the generation of research highlights using various sections of a research paper as input instead of abstract only. For this purpose, we employ a pointer-generator network with a coverage mechanism and pre-trained ELMo contextual embeddings [143] to generate the highlights. Unlike large pre-trained language models (often called foundation models [229]) that require access to a huge document corpus, large training time, a huge energy expenditure, our proposed method is task-specific, utilizes pre-trained embeddings, and is trained with a much smaller domain-specific corpus. We examined how well the proposed model can generate research highlights using two different types of input: (a) only the abstract, and (b) a combination of the abstract, introduction, and conclusion of a research paper.

This section is organized into four major sub-sections: (i) methodology, (ii) experimental setup, (iii) evaluation results, and (iv) analysis with a case study.

3.7.1 Methodology

In this section, we have described the model we proposed to generate research highlights from the scientific papers. We incorporated ELMo embeddings layer to pointer-generator network with coverage mechanism [174]. The workflow of our system is shown in Figure 3.3.

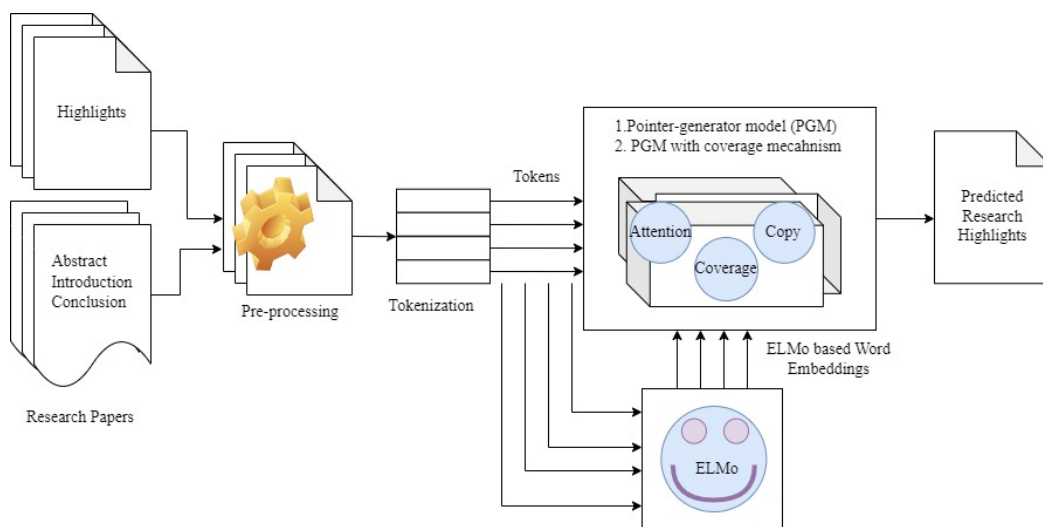


FIGURE 3.3: ELMo based encoder-decoder model for research highlight generation.

3.7.1.1 ELMo Pre-Trained Word Representations

Pre-trained word representations [141] have a significant role in many neural language understanding models. Pre-trained word representations are a key component in many neural language understanding models. On the other hand representation of learning with high quality and accurate generation are very challenging. A deep contextualized word representation, ELMo [143] can be used to capture the complex behaviour of word syntax, semantics and linguistic context. ELMo employs vectors that was trained with a stacked bidirectional LSTM.

3.7.1.2 Pointer-Generator Model with ELMo Embeddings and Coverage Mechanism

We use a pointer-generator model, proposed by See, Liu and Manning [174]. It consists of a seq2seq model with a BiLSTM encoder and an LSTM decoder with attention [129]. However, instead of using word embeddings trained from scratch or non-contextual embeddings like word2vec [125] or GloVe [142], we use context-sensitive embeddings that represent homonyms with different vectors, despite the fact that these words have the same spelling. In other words, word representations capture the fine differences in meaning that arise from the context in which the words are used. In our present work, we add a pre-trained ELMo contextual embeddings [143] layer that generates embedding for each word of the input text. Instead of directly passing the individual token id to the encoder recurrent neural network, we feed the token embeddings prepared by pre-trained ELMo [143] embeddings layer. This can improve the model's ability to generate hidden states because the input words embedding matrix is initialized with the pre-trained word embeddings ELMo. The embeddings are fine-tuned during model training. The dimension of ELMo word embeddings used in our experiment is 1024. The decoder has a unique copying technique, which decides between *copying* a word from the source text by utilizing copying mechanism or *generating* new words from the vocabulary (built from the vocabulary of the whole training corpus and the current input document). The copying mechanism helps to deal with out-of-vocabulary (OOV) words. The generating mechanism, on the other hand, induces new words which indicate novel paraphrasing. The decoder strikes a balance between copying words and generating words using a hyperparameter, which probabilistically chooses between the two alternatives. However, the pointer-generator model sometimes generates the same words repetitively. To overcome this problem, we used the coverage mechanism of Tu et al.[191]. In essence, this model focuses on the preceding time steps of the decoder through attention so that attending to the same word in the input document again and again is penalized.

3.7.2 Experimental Setup

In this sub-section, we discussed the datasets used, the data pre-processing steps, and the implementation details.

3.7.2.1 Dataset Details

We used the same dataset as defined in the previous work 3.6, and subsection 3.6.2.1, the CSPubSum dataset. First we crawl it. The following fields are typically present in the documents: title, abstract, author-written research highlights, authors-written keywords, introduction, related work, experiment, conclusion, and other major sub-sections that are part of the discourse structure of a research paper. For our experiments, we divided the dataset into training, validation, and test subsets (train, val, test) in proportion of 80 : 10 : 10 as same as previous work 3.6.

3.7.2.2 Data Pre-Processing

Before inputting the dataset to the model, we did some basic pre-processing steps. We removed unintended symbols, letters, urls, HTML tags and special characters as same as previous work 3.6, and subsection 3.6.2.2. Then we changed the dataset to lowercase. To conduct experiments, we rearranged the dataset in several ways. In particular, we organized it as (*abstract, research highlights written by author*), (*abstract \oplus introduction \oplus conclusion, research highlights written by author*), where text concatenation is represented as ' \oplus '. When considering only abstract as an input, we allowed a maximum of 400 tokens. In the case of combined inputs from abstract, introduction and conclusion sections, we allowed a maximum of 1500 tokens. From each section, we allowed up to 500 tokens. In all cases, the token count of model-generated research highlights was limited to 100 only.

3.7.2.3 Implementation Details

We conduct experiments with four different variations: (1) Pointer-generator model proposed by [174] (**PGM**), (2) Incorporating coverage mechanism (proposed in [191]) into the pointer-generator model (the combined model is also referred to in the same work [174]) (**PGM + Coverage**), (3) Pre-trained ELMo embeddings [143] with pointer-generator model (**PGM + ELMo**), and (4) Pre-trained ELMo embeddings with pointer-generator model and coverage mechanism (**PGM + ELMo + Coverage**).

All the models were trained on the GPU-supported Colab Pro+ environment. The pointer-generator network with ELMo embeddings used 1024 as the word embedding

TABLE 3.2: Evaluation of pointer-generator type models with and without ELMo embeddings: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on different inputs from the CSPubSum dataset. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. All scores are presented as percentages (%).

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-S	ROUGE-SU	METEOR	MoverScore	BERTScore
abstract only	PGM	35.44	11.57	29.88	11.45	12.35	25.4	56.69	83.80
	PGM + Coverage	36.57	12.3	30.69	12.14	13.04	25.4	57	84.05
	PGM + ELMo	35.74	12.54	32.47	11.81	12.71	19.75	54.98	82.62
	PGM + ELMo + Coverage	38.4	13.32	35.45	13.41	14.35	30.61	57.94	86.65
abstract + introduction + conclusion	PGM	33.49	10.83	30.87	10.67	11.59	25.51	56.75	86.01
	PGM + Coverage	35.73	11.61	32.96	11.6	12.52	27.71	57.39	86.26
	PGM + ELMo	33.6	11.44	30.97	10.78	11.69	25.68	56.79	86.02
	PGM + ELMo + Coverage	36.34	12.11	33.77	11.98	12.97	27.78	57.63	86.68

dimension and that without ELMo embeddings used 128 as the word embedding dimension. We used mini-batches of size 16. For all models, the maximum vocabulary size was restricted to 50K tokens. For all models, the dimension of RNN hidden states is 256. We chose maximum gradient norm of 1.2 for gradient clipping with other hyperparameters following [174].

3.7.3 Results

In this section, we compare scores from four model variants: (1) Pointer-generator model (*PGM*), (2) Pointer-generator model with coverage (*PGM + Coverage*), (3) Pointer-generator model with ELMo embeddings (*PGM + ELMo*), and (4) Pointer-generator model with ELMo embeddings and coverage (*PGM + ELMo + Coverage*). Models (1) and (2) use randomly initialized word embeddings. Our proposed model is (4), with the other variants serving as an ablation study. Table 3.2 shows that (**PGM + ELMo + Coverage**) consistently outperforms the others across all metrics, whether using only the abstract or a combination of abstract, introduction, and conclusion as input.

3.7.4 Case Studies

In this sub-section, we have shown a few research highlights generated by our models to enable a qualitative study of the performance of the models. In all case studies, yellow color represents **factual errors**, orange color represents **repeating phrases** and green color identifies some correctly **added words or phrases**.

Figure 3.4 shows the comparison of predicted research highlights generated by variants of pointer-generator models when the input (for training and test) is the abstract of a research paper. Observe that the first model *PGM* generates repeating phrase “total route duration”, which is solved when we add coverage mechanism. Similarly, a full sentence is repeated by the (*PGM + ELMo*) model which the fourth model (*PGM + ELMo + Coverage*) corrects using the coverage mechanism. Note that the words ‘sub-tour elimination constraints’ that the last model (*PGM + ELMo + Coverage*) generates

is present in the abstract of the paper, and its insertion in the output is semantically correct, although it is absent in the golden set of research highlights submitted by the authors. Another observation is that the models with **ELMo** embeddings display better linguistic quality with respect to grammatical syntax probably due to the contextual nature of the embeddings. For example, while the first model (*PGM*) contains a grammatically incorrect sentence like “We study minimize the ...” and the second model (*PGM + Coverage*) generates the incorrect sentence “Results are conducted ...”, the ELMo-based models do not display such issues. However, none of the models capture the last line “Comparative computational results indicate that a flow based formulation is superior the other three” of the highlights penned by the authors because it does not appear in the abstract.

Figure 3.5 depicts a similar comparison among the outputs of the four models for a different paper. Again, we notice that without the coverage mechanism, words are incorrectly repeated, while the coverage mechanism reduces repetition significantly. Grammatical correctness of ELMo-based models is also more than that of other models. Figure 3.6 shows the comparison of predicted research highlights generated by the models for the same paper when the input (for training and test) is the combination of a research paper’s abstract, introduction, and conclusion. Observe the same phenomenon of repetitive words in absence of the coverage mechanism: (*PGM + ELMo*) repeats the word ‘graphical’ several times, which is fixed when the coverage mechanism is added. However, a careful comparison of the outputs of (*PGM + ELMo + Coverage*) across Figures 3.5 and 3.6 show that the highlights produced in the first case are more meaningful. For example, the statement “We modeled the image as a superpixel based markov random field segmentation framework” in (*PGM + ELMo + Coverage*) model in 3.6) is not semantically correct. It seems that using a larger input from the body of the paper, rather than only the abstract, confuses the model, resulting in lower-quality output. Recollect that the superior performance of abstract-only models is quantitatively communicated by Table 3.2, and we now see a specific qualitative example of the performance difference. None of the models, irrespective of the input type, includes the information in the sentence “A copy of source codes will be released as an open source project for continuing studies.” mentioned as the last line of the author-written highlights.

Thus, in all cases, we observe that the research highlights generated by the pointer-generator with ELMo embeddings and the coverage mechanism with the abstract as input are semantically more meaningful and contains fewer syntactic issues compared to the outputs in other cases. However, we also note that the research highlights produced by the models do not agree perfectly with the author-written highlights when the latter contain information not included in the inputs to the model.

Author-written research highlights: “We introduce and model the traveling salesman problem with time dependent service times. The proposed model can handle several types of service time functions such as linear and quadratic. We describe properties of the service time function and compute valid lower and upper bounds. We propose and compare several mathematical programming formulations for the problem. Comparative computational results indicate that a flow based formulation is superior the other three.”
PGM: “We study minimize the total route duration of the total route duration . Consider the total route duration of the total time [UNK] describe basic properties for certain classes of service time functions. Model can handle several types of service time functions e.g. linear and quadratic functions. Examples obtained by valid lower and upper bounds are presented. Results obtained by implementing different quadratic service time functions are discussed.”
PGM + Coverage: “We study the route duration of the total route duration of total service time. New model is based on various linear and valid lower heuristics. Results are conducted to illustrate different linear and quadratic service time functions. Results obtained by implementing linear and quadratic functions are presented.”
PGM + ELMo: “A version of the classical traveling salesman problem with time dependent service times is proposed . The proposed model can handle several types of service time functions e.g. linear and quadratic functions. The proposed model can handle several types of service time functions e.g. linear and quadratic functions. Numerical results obtained by implementing different linear service time functions are presented.”
PGM + ELMo + Coverage: “This paper introduces a version of the classical traveling salesman problem with time dependent service times. The proposed model can minimize the total route duration which consists of the total service time. We apply several classes of subtour elimination constraints and measure their effect on the performance of our model . Numerical results obtained by implementing different test instances are presented.”

FIGURE 3.4: The input consists only of the abstract of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without ELMo embeddings are shown. The input abstract and the author-written research highlights are taken from <https://www.sciencedirect.com/science/article/pii/S037722171500702X>.

3.7.5 Summary of Findings

This work, explores generating research highlights using a pointer-generator network with coverage and ELMo embeddings, incorporating paper multiple sections for richer input. Unlike in the previous section 3.6, which used GloVe embeddings and focused only on abstracts, this method combines multiple sections of a paper along with the abstract for richer input. Unlike large pre-trained models, it is task-specific, uses pre-trained embeddings, and requires a smaller, domain-specific corpus. Evaluation with ROUGE, METEOR, MoverScore, and BERTScore shows superior performance, aiming to enhance the clarity of scientific literature.

Author-written research highlights: “We proposed a novel gap search markov random field mrf for accurate cervical smear image segmentation. This method could acquire three regions nuclei cytoplasm and background automatically by a label map mechanism. The gap search algorithm is faster than other three algorithms in the experiments. A copy of source codes will be released as an open source project for continuing studies.”			
PGM: “A novel model is developed to classify		color non overlapping superpixel patches .	
Segmentation is used for the first time for image segmentation .		Gap search algorithm for one image segmentation and superpixel based algorithms. Results are much more faster than pixel based and superpixel based model. Algorithms are more faster than pixel of superpixel based algorithms.”	
PGM + Coverage: “ We seek to classify color non overlapping superpixel patches on one image image . Work presents an label map mechanism to acquire the whole image as the undirected probabilistic graphical model . Gap search algorithm was designed to enhance the model efficiency. Algorithms much more faster than pixel based and superpixel superpixel .”			
PGM + ELMo:		“We propose a novel superpixel based markov random field mrf segmentation framework . The model describes the nucleus cytoplasm and image background of cell images. A gap search algorithm is designed to solve the model efficiency. The gap search algorithm is much more faster than pixel based and superpixel based algorithms.”	
PGM + ELMo + Coverage:		“A novel image segmen- tation method for automated cervical cell analysis is proposed. The whole image as an undirected probabilistic graphical model is presented . A gap search method is proposed to solve the nuclear cytoplasmic and background regions . The proposed algorithm is much more faster than pixel based and superpixel based algorithms.”	

FIGURE 3.5: The input consists only of the abstract of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without ELMo embeddings are shown. The input paper is at <https://www.sciencedirect.com/science/article/pii/S0010482516300154>

3.8 Pointer-Generator Model with SciBERT Word Embeddings and Coverage Mechanism

In this work [164], we extract research highlights from scientific papers using abstractive summarization techniques. The work’s key contributions are: (1) We introduce a novel method for generating research highlights automatically. Our approach combines a SciBERT pre-trained word embedding layer with a pointer-generator network enhanced by a coverage mechanism. Unlike traditional models, we integrate SciBERT, a BERT variant trained on scientific literature, at the input stage to produce higher-quality abstractive summaries. This is the first work to use SciBERT within this framework for research highlight generation. We test our model on a benchmark dataset, CSPubSum. On the CSPubSum dataset, our model achieves the best performance when the input is only the abstract of a paper as opposed to other segments of the paper. It achieves ROUGE-1, ROUGE-2, and ROUGE-L F1 scores of 38.26, 14.26, and 35.51, respectively, a METEOR score of 32.62, a MoverScore F1 of 20.19, and a BERTScore F1 of 86.65, surpassing all other baseline. (2) We fine-tuned various pre-trained models from Hugging Face, such as T5-base, Distilbart-CNN-12-6, GPT-2, and ProphetNet-large-uncased-cnndm, on the CSPubSum dataset. We then compared their performance with our proposed model, focusing on both their summary generation capabilities and the resources required for training, with an emphasis on energy efficiency and carbon footprint.

This section is organized into four major sub-sections: (i) methodology, (ii) experimental setup, (iii) evaluation results, and (iv) analysis with a case study.

3.8.1 Methodology

In this section, we have described the model we use to generate research highlights from the scientific documents.

We use pointer-generator networks to produce highlights from scientific papers. It consists of a seq2seq model with a BiLSTM encoder and an LSTM decoder with attention [129]. However, while the original model proposed by See, Liu, and Manning [174] uses word-embeddings – they are learned from scratch during training – we used a pre-trained transformer to generate the contextual embeddings of the tokens in the input document. The architecture of our model is shown in Figure 3.7. Proposed model: Pointer-generator network with coverage mechanism and SciBERT word embeddings.

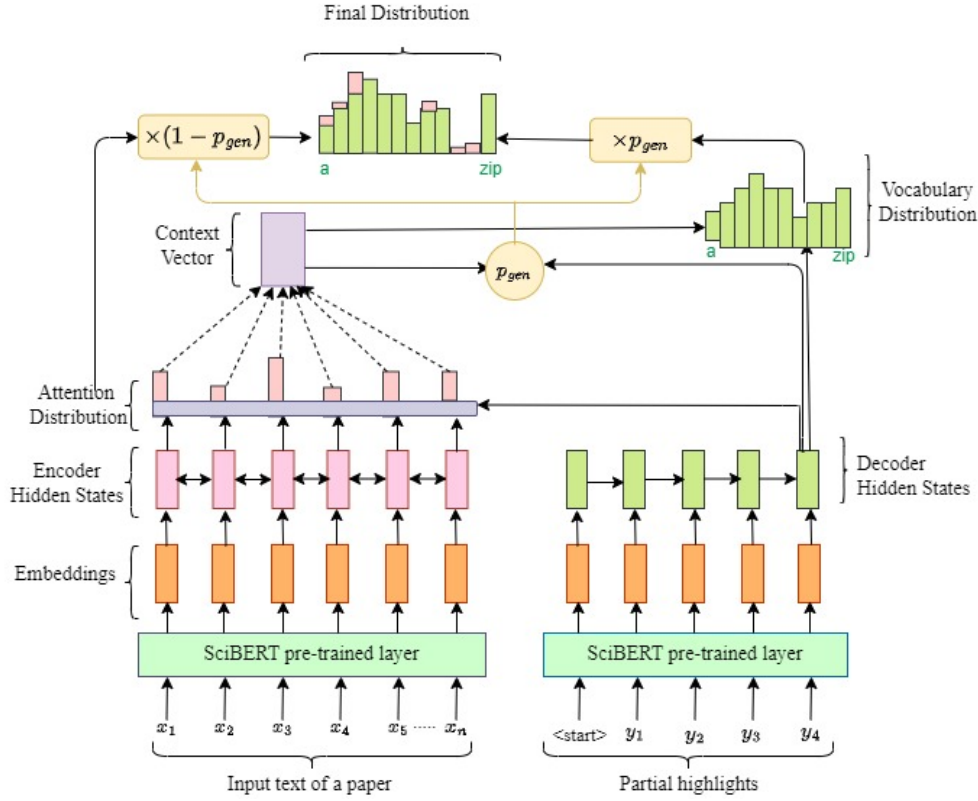


FIGURE 3.7: Proposed model: Pointer-generator network with coverage mechanism and SciBERT word embeddings.

3.8.1.1 BERT and SciBERT

The pre-trained language model BERT stands for *bidirectional encoder representations from transformers*. BERT has been pre-trained on the tasks of masked language modelling (MLM) and next sentence prediction (NSP) [50]. Normally, standard conditional language models are trained on either left-to-right or right-to-left representations of the context, but MLM used both left-to-right and right-to-left representations of the context. The primary goal of the masked language model is to predict the actual vocabulary identifier of the input's randomly masked tokens. Next sentence prediction (NSP) aids the model in comprehending sentence relationships. This feature helps to improve the performance for the downstream tasks of question-answering (QA) and natural language inference (NLI). To encode the input, the input sentence is first tokenized, and then the tokens are combined with 3 new tokens, namely, CLS, SEP, and MASK. CLS is added at the start of sentence to represent sentence-level classification. To predict the next sentence, SEP is used. During the MLM task, MASK is used to represent masked tokens. English Wikipedia (2,500M words) and the BooksCorpus (800M words) are used for pre-training the BERT model. Summing the corresponding token, segment,

and position embeddings yields the input representation for a given token. Primarily, BERT has two variants named as BERT_{BASE} and BERT_{LARGE}. BERT_{BASE} has 12 transformer layers, 768 hidden size, 12 attention heads, and 110M total parameters. BERT_{LARGE} has 24 transformer layers, 1024 hidden size, 16 attention heads and 340M total parameters.

SciBERT is a BERT-based pre-trained language model that was trained on a large corpus of scientific text from Semantic Scholar [10]. The same size and configuration of BERT_{BASE} is used to train the SciBERT model and allowed 128 tokens of maximum sentence length. SciBERT has 4 variants: **cased/uncased** and **basevocab/scivocab**. The **basevocab** models are fine-tuned from the corresponding BERT_{BASE} models. The **scivocab** models have been trained from scratch.

3.8.1.2 Pointer-Generator Model with SciBERT Embeddings

This model consists of a word-embedding layer and a pointer-generator network. The word-embedding layer converts the words in the input document to embeddings. We have used a pre-trained SciBERT model [21] to generate word embeddings. Using this mechanism, each word (x_t) in the encoder and the decoder part will be represented as an embedding vector \vec{x}_t as:

$$\vec{x}_t = g(x_t) \quad (3.5)$$

where $g(.)$ is the embedding-generating function. The CLS token has been added to represent sentence-level classification. Here, the main use of SciBERT [21] is that instead of directly feeding the token ids of the input document into the encoder, we are passing the pre-trained SciBERT-generated word embeddings. In our experiments, the dimension of word embeddings is 768. A pointer-generator network [174] augments the sequence-to-sequence (seq2seq) model with attention [129] using a special copying mechanism. When generating words, the decoder probabilistically decides between generating new words from the vocabulary (i.e., from the training corpus) and copying words from the input document (by sampling from the attention distribution). While the generator helps in novel paraphrasing, copying helps to tackle OOV words. This improves the model’s ability to calculate hidden states because the inputs at each time step have been accurately and completely represented, contributing to the improvement of the attention distribution. At each decoder time step t , the probability of generating a new word is

$$P_{gen} = \sigma(\vec{W}_{h^*}^\top \vec{h}_t^* + \vec{W}_s^\top \vec{s}_t + \vec{W}_x^\top \vec{x}_t + \vec{b}_{ptr}) \quad (3.6)$$

where \vec{h}_t^* is the context vector, \vec{s}_t is the decoder hidden state, \vec{x}_t is the decoder input (which is the decoder output at time $t - 1$ during test, and the correct word at time $t - 1$

during training), σ is the *sigmoid function*, and \vec{W}_{h^*} , \vec{W}_s , \vec{W}_x and \vec{b}_{ptr} are the learnable parameters. Hence, for the SciBERT pre-trained embeddings layer the formula in (3.6) is modified as follows:

$$P_{gen} = \sigma(\vec{W}_{h^*}^\top \vec{h}_t^* + \vec{W}_s^\top \vec{s}_t + \vec{W}_x^\top g(\mathbf{x}_t) + \vec{b}_{ptr}) \quad (3.7)$$

To predict the next word y_t , the probability distribution over the extended vocabulary (i.e., the fixed vocabulary of the training corpus and the present document) is calculated:

$$P(y_t) = P_{gen}P_{vocab}(y_t) + (1 - P_{gen}) \sum_{i:w_i=y_t} a_{t,i} \quad (3.8)$$

where \vec{a}_t is the attention distribution over the fixed vocabulary at time t , $a_{t,i}$ is the attention over the word w_i at time t , and P_{vocab} is the probability distribution over the extended vocabulary generated by the softmax layer of the decoder. The loss for decoder time step t is:

$$loss_t = -\log P(y_t^*) \quad (3.9)$$

where y_t^* is the target word. The overall loss for the sequence is the average of the losses over all the decoder time steps for this sequence.

3.8.1.3 Pointer-Generator Model with SciBERT Embeddings and Coverage Mechanism

Sometimes the above pointer-generator network redundantly generates the same word multiple times during test. The coverage model of Tu et al. [191] aims to address this problem. This model essentially gives attention to the previous timesteps of the decoder. It computes a coverage vector $\vec{c}^\#$ defined as the sum of the attention distributions \vec{a}_t over all previous timesteps $\tau = 1$ to $\tau = t - 1$ of the decoder:

$$\vec{c}_t = \sum_{\tau=0}^{t-1} \vec{a}_\tau \quad (3.10)$$

Note that \vec{c}_0 is a zero vector. The *coverage vector* will be taken as an extra input to the attention mechanism that is used by the decoder while generating the next word.

The *coverage loss* quantifies if the model is continuously giving more attention to the same words:

$$CoverageLoss_t = \sum_i \min(a_{t,i}, c_{t,i}) \quad (3.11)$$

Finally, the coverage loss is included in the primary loss function of the decoder. The

revised loss for decoder time step t can be written using a hyperparameter λ as follows:

$$loss_t = -\log P(y_t^*) + \lambda \sum_i \min(a_{t,i}, c_{t,i}) \quad (3.12)$$

3.8.2 Experimental Setup

In this sub-section, we discussed the datasets used, the data pre-processing steps, and the implementation details.

3.8.2.1 Dataset Details

We used the same dataset as in the previous two contributions in this thread, sections 3.6 and 3.7, namely the CSPubSum dataset, with the same 80:10:10 split for training, validation, and testing.

3.8.2.2 Data Pre-Processing

We have used the Stanford CoreNLP Tokenizer³ for tokenizing the sentences. The whole corpus is first converted to lowercase. We have removed all unnecessary symbols, letters, and other elements from the text that do not affect the aim of our research. In particular, HTML tags, parentheses, and special characters have been removed.

Then we reorganized the dataset in several ways to perform various experiments. We organized it as (*abstract, author-written research highlights*), (*conclusion, author-written research highlights*), (*introduction, author-written research highlights*), (*abstract* \oplus *conclusion, author-written research highlights*), and (*introduction* \oplus *conclusion, author-written research highlights*) where ‘ \oplus ’ denotes text concatenation. Since the background and a broad summary of the paper normally appear in the introduction, and the main findings of the paper are mentioned in the conclusion, we experiment taking these sections as inputs. Since an overview of the paper is present both in the introduction and the abstract, we do not use them together, rather we use the combinations (abstract \oplus conclusion), and (introduction \oplus conclusion).

When the abstract is used as the input, we set the maximum number of input tokens to 400. When the conclusion is used as the input, the maximum number of input tokens allowed is 800. When the introduction is used as the input, the maximum number of input tokens allowed is set to 1200. For all other inputs, we have restricted the input size to 1500 tokens. In all cases, the maximum token count of the generated research highlights tokens is set to 100. The above figures are motivated by the observation that the average length of an abstract is 186, the average length of the author-written

³<https://stanfordnlp.github.io/CoreNLP/>

highlights in a paper is 52, and the average length of the conclusion is 425, that of the introduction is 837, the average length of (abstract \oplus conclusion) is 643, and that of (introduction \oplus conclusion) is 1230.

3.8.2.3 Implementation Details

We trained four variants of the proposed model: pointer-generator network with word embeddings trained from scratch as part of the model training (**PGM**), pointer-generator network with coverage mechanism where word embeddings are trained from scratch as part of the model training (**PGM + Coverage**), pointer-generator network with SciBERT embeddings for the input tokens (**PGM + SciBERT**), and pointer-generator network with coverage mechanism and SciBERT embeddings for the input tokens (**PGM + SciBERT + Coverage**). For all variants of SciBERT models, during model training, the embeddings are fine-tuned. We trained all models on Tesla P100-PCIE-16GB Colab Pro+ that supports GPU-based training. We used mini-batches of size 16. For all models, we used bidirectional LSTMs with cell size of 256. For models without SciBERT, word embeddings of dimension 128 are trained end-to-end with the model. For models with SciBERT, pre-trained word embeddings of dimension 768 are used. For all experiments, we constrained the vocabulary size to the most frequent 50,000 tokens. We considered gradient clipping with a maximum gradient norm of 1.2. Out of the four variations of the SciBERT model, we use **SciVocab-uncased**⁴. We used other hyperparameters as suggested by [174]. We have used the validation set to determine the number of epochs for training.

3.8.3 Results

In this sub-section, we report the results of experiments on the CSPubSum dataset for various input types.

Input: Abstract: Results are shown in Table 3.3 for ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore, and BERTScore when the input is the abstract of a research paper. We observe that among the four models, the pointer-generator network with coverage mechanism and SciBERT (**PGM + SciBERT + Coverage**) model achieve the highest ROUGE, METEOR, MoverScore, and BERTScore values.

Input: Conclusion: Results are shown in Table 3.3 for ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore, and BERTScore when the input is only the conclusion of a research paper. We observe that among the four models, the (**PGM + SciBERT + Coverage**) model achieves the highest ROUGE, METEOR and BERTScore values.

Input: Introduction: Results are shown in Table 3.3 for ROUGE-1, ROUGE-2,

⁴https://huggingface.co/allenai/scibert_scivocab_uncased/

TABLE 3.3: Evaluation of pointer-generator type models with and without SciBERT embeddings: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on various inputs from CSPubSum dataset. All ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script. All scores are presented as percentages (%).

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore
abstract only	PGM	35.44	11.57	29.88	25.4	18.47	83.80
	PGM + Coverage	36.57	12.3	30.69	25.4	19.67	84.05
	PGM + SciBERT	36.55	13.44	33.57	30.34	19.15	86.34
	PGM + SciBERT + Coverage	38.26	14.26	35.51	32.62	20.19	86.65
conclusion only	PGM	32.11	9.32	29.62	24.04	14.78	85.72
	PGM + Coverage	34.33	9.73	31.71	24.99	18.09	86.07
	PGM + SciBERT	33.19	9.8	30.49	24.26	15.9	86.03
	PGM + SciBERT + Coverage	34.81	10.02	32.31	25.21	18.99	86.52
introduction only	PGM	30.85	7.92	28.55	19.76	14.87	85.25
	PGM + Coverage	32.46	8.18	30	20.50	17.21	85.48
	PGM + SciBERT	31.56	8.79	29.18	23.09	15.27	85.93
	PGM + SciBERT + Coverage	33.33	9.7	30.86	24.10	17.49	86.17
abstract + conclusion	PGM	29.85	8.16	25.80	19.38	14.18	83.19
	PGM + Coverage	31.70	8.31	26.72	20.92	15.73	83.49
	PGM + SciBERT	32.84	9.86	30.34	24.59	16.01	86.13
	PGM + SciBERT + Coverage	35.09	10.94	32.69	27.31	19.27	86.52
introduction + conclusion	PGM	29.78	7.47	25.15	19.25	14.23	83.05
	PGM + Coverage	31.63	7.65	26.25	20.24	15.76	83.32
	PGM + SciBERT	32.38	9.63	29.79	23.95	16.16	86.11
	PGM + SciBERT + Coverage	35.32	10.93	32.76	26.57	19.39	86.59

ROUGE-L, METEOR, MoverScore, and BERTScore when the input is the introduction of a research paper. We observe that among the four models, the (**PGM + SciBERT + Coverage**) model achieves the highest ROUGE, METEOR, MoverScore, and BERTScore values.

Input: Abstract + Conclusion: Results are shown in Table 3.3 for ROUGE-1, ROUGE-2, ROUGE-L scores, METEOR, MoverScore, and BERTScore when the input is the combination of the abstract and the conclusion of a paper. We again observe that the best performance is achieved by the (**PGM + SciBERT + Coverage**) model.

Input: Introduction + Conclusion: When the inputs is the combination of introduction and conclusion in the test dataset, we record ROUGE-1, ROUGE-2, ROUGE-L scores, METEOR, MoverScore, and BERTScore as shown in Table 3.3. The best performing model is (PGM + SciBERT + Coverage). Upon analysis of the dataset, we found that in many cases the highlights are largely included in the ‘abstract’; therefore, using the ‘abstract’ as input to the model results in high performance. We have observed that the ‘conclusion’ typically presents a more detailed and technically dense description of the findings in contrast to the more overview-style summary included in the research highlights (see, for example, these papers^{5 6}). The ‘conclusion’ also includes future work, which does not form part of the highlights. So adding the ‘conclusion’ with the ‘abstract’ does not improve the performance. Although the ‘introduction’ of

⁵<https://www.sciencedirect.com/science/article/abs/pii/S0010448514001870>

⁶<https://www.sciencedirect.com/science/article/pii/S0010448514001638>

a paper often contains the main findings of the paper, it also contains a lot of other information (typically, to build the background and context to the current work) that is not included in the highlights and must be filtered away by the model when generating the output.

3.8.3.1 K -Fold Cross-Validation

We also perform K -fold cross-validation (CV) of our model ($PGM + SciBERT + Coverage$) on the CSPubSum dataset. For this purpose, we set $K = 5$, that is, we split the whole dataset into five distinct parts. We trained using four parts (or folds) and tested the model using the remaining part. In each case, we trained the pointer-generator network with SciBERT for 20000 iterations, then added the coverage mechanism and continued training for another 1000 iterations. In all cases, we consider only the abstracts of the CSPubSum dataset as the input. Table 3.4 reports the ROUGE, METEOR and BERTScore for the model ($PGM + SciBERT + Coverage$) with 5-fold cross-validation and compares the performance with that of holdout validation. Since K -fold cross-validation is computationally quite expensive, we did not conduct it for the other input types. Note that the performance achieved by K -fold cross-validation is slightly higher than that reported by holdout validation. Since it is widely believed (see, for example, [88, 156]) that K -fold cross-validation results are a better indicator of the generalization performance, our model is likely to be better than that indicated by holdout testing.

TABLE 3.4: K -fold cross-validation of the pointer-generator type models with and without SciBERT embeddings on the CSPubSum dataset. For comparison, the performance of the models with holdout validation are reproduced from Table 3.3.

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore
abstract only (holdout validation)	PGM	35.44	11.57	29.88	25.4	18.47	83.80
	PGM + Coverage	36.57	12.3	30.69	25.4	19.67	84.05
	PGM + SciBERT	36.55	13.44	33.57	30.34	19.15	86.34
	PGM + SciBERT + Coverage	38.26	14.26	35.51	32.62	20.19	86.65
abstract only (5-fold CV)	PGM + SciBERT	37.79	12.77	34.78	29.92	19.87	86.72
	PGM + SciBERT + Coverage	39.43	15.25	36.48	30.85	22.23	87.01

3.8.3.2 Comparison with Pre-Trained Models

We have chosen the following pre-trained models from the Hugging Face website for the purpose of comparison: **T5-base**⁷, **Distilbart-CNN-12-6**⁸, **GPT-2**⁹ and **ProphetNet-large-uncased-cnndm**¹⁰. We fine-tuned all four models to 15 epochs with CSPubSum

⁷<https://huggingface.co/t5-base>

⁸<https://huggingface.co/sshleifer/distilbart-cnn-12-6>

⁹<https://huggingface.co/gpt2>

¹⁰<https://huggingface.co/microsoft/prophetnet-large-uncased-cnndm>

where 8115 documents (each comprising an abstract and author written research highlights) are taken for training. We tested them on the test dataset of 1013 examples. We used a batch size of 4 for fine-tuning all four pre-trained models. Observations on the test set are shown in Table 3.5. The performance of ProphetNet-large-uncased-cnndm pre-trained model is significantly worse than that of other models; the training duration and compute resources we used appeared to be inadequate for this model. We observe that T5-base performs better than the other models in terms of ROUGE and BERTScore metrics while Distilbart-CNN-12-6 gives the highest METEOR score. The slight performance gain of pre-trained models is not surprising at all given the number of parameters and the exhaustiveness of the training of such models. Rather the closeness of the proposed model, which does not require fine-tuning a large pre-trained transformer model, appears to demand more attention to strike the right trade-off between performance and the resources needed for training.

In the next subsection, we will discuss an important aspect of these large models, which has received attention in the recent years. This aspect deals with the energy efficiency of algorithms that is also related to the consequent carbon footprint.

TABLE 3.5: Performance of fine-tuned versions of pre-trained models on the CSPubSum dataset using abstracts of the papers as the input. All metrics (ROUGE-1, ROUGE-2, ROUGE-L, METEOR, and BERTScore) are reported as F1-scores and presented as percentages (%). The highest scores are highlighted in bold.

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	BERTScore
T5-base	40.03	16.27	37.64	36.33	86.80
Distilbart-CNN-12-6	39.95	16.13	37.16	38.99	86.69
GPT-2	33.12	11.76	30.64	33.14	85.30
ProphetNet-large-uncased-cnndm	23.95	0.96	20.38	15.3	81.41
PGM + SciBERT + Coverage (ours)	38.26	14.26	35.51	32.62	86.65

3.8.3.3 Analysis of Energy Consumption

Recently transformer architectures have significantly improved the performance of various natural language processing (NLP) tasks. Inspired by the original transformer [195], language models such as ELMo [143], BERT [50], GPT family [152] and BART [98] have emerged and produced state-of-the-art performance on various tasks. However, they require enormous amounts of data and compute resources for pre-training. This large computation consumes a lot of energy and has a high carbon footprint. It has an adverse financial and environmental impact [182, 95].

The expression to calculate carbon footprint C (in gram carbon dioxide equivalent or gCO₂e) as given in the equation 3.13 is taken from [95].

$$C = t \times (n_c \times P_c \times u_c + n_m \times P_m) \times PUE \times CI \times 0.001 \quad (3.13)$$

We modified Equation 3.13 to Equation 3.14:

$$C = t \times (n_c \times P_c \times u_c + n_{gpu} \times P_{gpu} \times u_{gpu} + n_m \times P_m) \times PUE \times CI \times 0.001 \quad (3.14)$$

where t is the running time (in hours), n_c is the number of cores, P_c is the power draw of a computing core, u_c is the core usage factor (between 0 to 1), n_{gpu} is the number of GPUs, P_{gpu} is the power drawn by the GPU, u_{gpu} is the GPU usage factor (between 0 to 1), P_m is the power draw of a memory unit (in watt). The power draw of memory is considered as 0.3725 W per GB [82, 95].

We trained all the models on Tesla P100-PCIE Colab Pro+ that supports GPU. The efficiency coefficient of the data center is known as PUE (power usage effectiveness). Google uses ML to reduce its global yearly average PUE to 1.10 [62]. We use average worldwide value as carbon intensity (CI) of 475 gCO₂e KW/hour[77]. Gross CO₂ emission during training for T5 pre-trained model [154] was 46.7 tCO₂e [139], any transformer_{big} model training required 192 lbsCO₂e [182] and BERT base model with GPU required 1438 lbsCO₂e [182]. We measure memory and compute power consumption and emission of CO₂ footprint using the WandB tool ¹¹. The quantitative results are shown in Table 3.6. In our proposed model, we require SciBERT embeddings of the input documents as input. So as a pre-processing step before model training, we encode the documents with SciBERT: this is a one-time operation and not repeated in every epoch. Table 3.6 clearly shows that our proposed model (third column) has fewer trainable parameters, and lower computational overhead and smaller carbon footprint per epoch than those of the other models. We have graphically compared the % of GPU utilization, % of CPU utilization, GPU Power usage, GPU memory allocated, memory used by process and required process CPU threads of the models over the training duration in Figure 3.8. The figure shows that GPU and CPU utilization, GPU power usage, and the process memory used by our proposed model are lower than those used in fine-tuning the large pre-trained summarization models. While our model consumes a large memory for a short time, the other models typically have a larger memory consumption that remains steady for a longer duration. Our model exploits more CPU threads than GPT-2 but fewer threads than other compared models. We believe that researchers should give attention to energy-friendly models and algorithms rather than only to performance metrics. In this context, our model is a better alternative to large

¹¹<https://wandb.ai/site>

pre-trained transformers.

TABLE 3.6: Power consumption, compute expenditure, and CO₂ emission statistics for summarization models.

Factors	Sub-Factor	PGM + SciBERT + Cover- age	ProphetNet large- uncased- cnndm	GPT-2	Distilbart- CNN- 12-6	T5- base
Total trainable parameters		21.5M	391M	117M	305M	220M
Colab Notebook	Avail. RAM: 51GB	2.86GB	4.07GB	3.20GB	6.47GB	3.89GB
	Avail. GPU: 16GB	1.14GB	15.57GB	13.51GB	14.64GB	2.92GB
	Avail. Disk: 166.83GB	48.30GB	41.53GB	40.64GB	41.21GB	44.57GB
Power consumed	Max. GPU power: 250W	116W	189W	187W	172W	206W
	Max. CPU power: 95W	95W	95W	95W	95W	95W
% of GPU utilization		75%	100%	97%	100%	97%
% of GPU memory allocated		55%	99%	89%	92%	60%
% of CPU utilization		35%	16%	27%	19%	17%
Used process CPU threads		54	57	38	55	66
Process memory in use (GB)		0.834	22.28	3.84	19.18	12.42
Time for one epoch (mins)		5.17	31	22	19	15
Epoch-wise carbon footprint (gms/epoch)		5.56	56.72	40.68	32.35	28.93

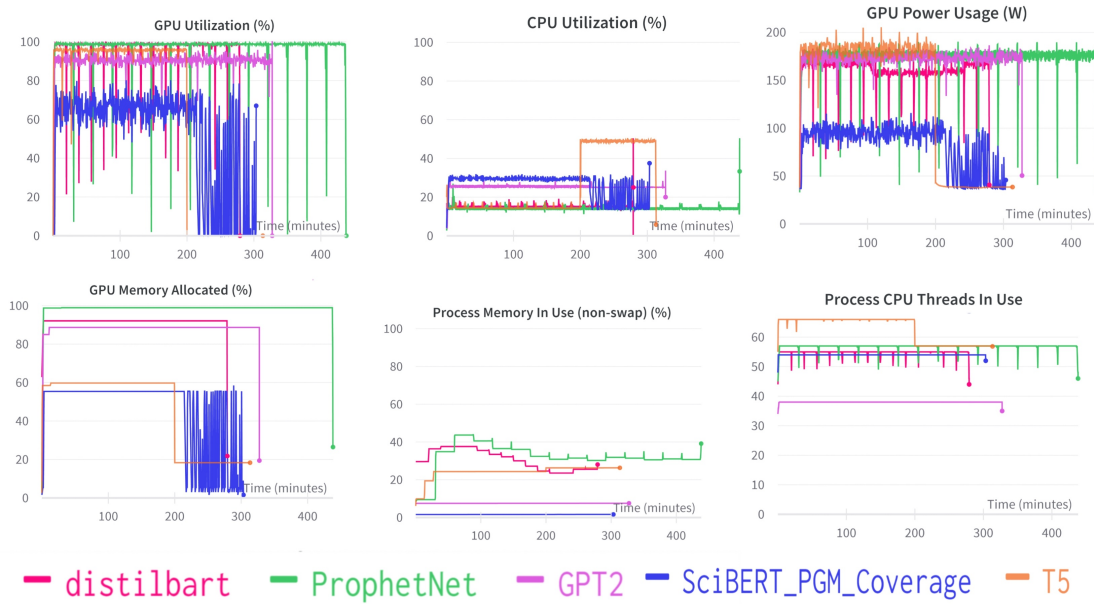


FIGURE 3.8: Comparison of compute resources used by summarization models.

3.8.4 Case Studies

We now present a few examples demonstrating the outputs produced by the pointer-generator type models used in this paper. In all the case studies reported below, *yellow* color represents **factual errors** and *orange* shows **repeating words**. Figure 3.9 illustrates the highlights produced by the four models when the input is only the abstract. Note that the vanilla pointer-generator network misses or incorrectly uses some keywords while generating the highlights. For example, it produces “... algorithm for expression data clustering.” instead of “... algorithm for microarray gene expression data clustering.”, and “... type 2 fuzzy means” instead of “... type 2 fuzzy c-means”. While use of SciBERT corrects these issues, unnecessarily repeated words are seen when coverage mechanism is absent. The output produced by the (*PGM + SciBERT + Coverage*) model is closest to the author-written highlights. Figure 3.10 and Figure 3.11 show the highlights produced by the four models when the input is only the conclusion and only the introduction, respectively. We make similar observations as above. Figure 3.12 and Figure 3.13 depict the highlights produced by the models when the input is (abstract + conclusion) and (introduction + conclusion), respectively. We observe that the highlights produced by all the models for the last four input types i.e (introduction + conclusion) contain a number of acronyms like ‘fcm’ (fuzzy C-means), ‘gt2’ (general type 2), ‘fss’ (fuzzy sets), and ‘cvi’ (cluster validity index) which occur frequently in the introduction and conclusion of the paper. Since the abstract typically does not contain acronyms, highlights generated using it are also generally free of acronyms.

3.8.5 Summary of Findings

For this work [164], we applied four different deep neural models to generate research highlights from a research paper. We experimented with different input types for each model for the CSPubSum dataset. The pointer-generator model with SciBERT and coverage mechanism (**PGM + SciBERT + Coverage**) achieved the best performance in each case. But the predicted research highlights are not yet perfect in terms of syntax and semantics. We are currently exploring other techniques to address these issues. A few other research directions would be to generate highlights that summarize a set of related papers, and to build a database containing research findings from different papers with links connecting semantically-related findings.

<p>Author-written research highlights: “Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.”</p>
<p>PGM: “We propose a new two stage meta heuristic algorithm for expression data clustering. New objective function for general type 2 fuzzy means is proposed. Output is inserted to the first stage until no better solution is obtained. Proposed approach is evaluated using three microarray gene expression datasets. Experiments verify the performance of the proposed approach.”</p>
<p>PGM + Coverage: “A novel meta heuristic methodology for microarray gene expression data clustering is proposed. Two stage inspired the two stage meta heuristic optimization framework simulated annealing is proposed. Philosophy of the meta heuristic framework is devoted to the annealing process accompanied by its proposed perturbation mechanisms. Experiments were conducted using three microarray gene expression datasets. Experiments demonstrate the capabilities of the proposed approach.”</p>
<p>PGM + SciBERT: “A new two stage meta heuristic algorithm is proposed. A new objective function using planes for general type 2 fuzzy sets is presented. A new objective function using planes for general type 2 fuzzy C means clustering algorithm is proposed. The proposed approach has been evaluated using several synthesized datasets and three microarray gene expression datasets. The proposed approach is evaluated using several synthesized datasets and three microarray gene expression datasets.”</p>
<p>PGM + SciBERT + Coverage : “A powerful data clustering approach based on the concept of planes in general type 2 fuzzy sets is presented. A new objective function using planes for general type 2 fuzzy c means clustering algorithm is proposed. The proposed approach has been evaluated using three microarray gene expression datasets.”</p>

FIGURE 3.9: Input is only the abstract from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from <https://www.sciencedirect.com/science/article/pii/S0010482514001565>.

<p>Author-written research highlights: “Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.”</p>
<p>PGM: “We propose a clustering algorithm based on the concept of planes in gt2 clusters. The proposed algorithm is based on simulated annealing and fcm approach. The framework is based on simulated annealing and fcm algorithm results. The proposed approach can be applied to the near optimal number of clusters. The proposed framework can be applied to the near optimal number of clusters.”</p>
<p>PGM + Coverage: “Clustering quality is a two stage sa clustering algorithm. Clusters of each cluster number is close to each other clusters. Spherical clusters at other clusters are considered. The number of clusters is raised to find near optimal number of clusters. An near optimal and optimal fuzzy sa can be described.”</p>
<p>PGM + SciBERT: “We propose a two stage sa clustering approaches for microarray gene expression. The gt2 fcm is repeated using a gt2 fuzzy cvi. The gt2 fuzzy sa will be detect the optimal number of clusters. The gt2 fcm is implemented using a gt2 fuzzy cvi.”</p>
<p>PGM + + Coverage + SciBERT: “The new fuzzy logic is applied to compute the similarity of two stage sa clustering. The gt2 fcm based clustering algorithms are used for different clusters. The results are analyzed using a gt2 fuzzy cvi algorithm. Better results are comparable to state of the art methods.”</p>

FIGURE 3.11: Input is only the introduction from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from <https://www.sciencedirect.com/science/article/pii/S0010482514001565>.

Author-written research highlights: “Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.”
PGM: “We propose a two stage sa clustering algorithm based on the concept of type 2 fcm and fcm [UNK] utilizes a two stage sa clustering algorithm based on the new simulated annealing. Present the first hybrid meta heuristic algorithm to solve the problem. Present the first stage solution approach to solve the problem.”
PGM + Coverage: “We present a two stage sa clustering algorithm based on the concept of planes in gt2 fss. Utilizes a two stage solution based on the new simulated annealing technology. Mechanism performs very complex times using the proposed framework. Results have been shown to be suitable for other systems.”
PGM + SciBERT: “A two stage sa clustering algorithm is proposed based on the concept of planes in gt2 fss. The objective function of the gt2 fuzzy sa will be performed. Quality of the proposed approach is analyzed using a gt2 fuzzy cvi fuzzy cvi. The proposed approach is one of the first stage of the proposed approach.”
PGM + SciBERT + Coverage : “A two stage sa clustering algorithm based on planes in gt2 fss is proposed. Only the optimal number of clusters are obtained. The mechanism of gt2 fuzzy sa will be used to detect the optimal number of clusters. The objective function of simulated annealing and fcm is further analyzed.”

FIGURE 3.12: Input is (abstract + conclusion) from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from <https://www.sciencedirect.com/science/article/pii/S0010482514001565>.

Author-written research highlights: “Presenting a new two stage meta heuristic clustering algorithm based on general type 2 fuzzy sets. Incorporating a new similarity based objective function using alpha plane representation of general type 2 fuzzy sets. Implementing the proposed approach on real microarray gene expression datasets.”
PGM: “A novel annealing clustering algorithm based on the sa clustering algorithm is proposed. Objective function of both planes is analyzed [UNK] on simulated annealing fcm is performed [UNK] on simulated clusters are provided.”
PGM + Coverage: “Automated annealing clustering algorithm is proposed for the stage problem of center splitting and center . Objective function of the fcm algorithms is implemented. Optimal number of clusters is gained using the first stage. Fuzzy algorithm of a gt2 cvi on the proposed approach is presented.”
PGM + SciBERT: “We proposed a two stage sa clustering algorithm based on the concept of planes in microarray gene expression data. We propose a two stage sa clustering framework to measure the similarity between members of each cluster. The proposed approach is primarily applied to the literature.”
PGM + SciBERT + Coverage : “We present a two stage sa clustering approaches for microarray gene expression data. the algorithm is implemented on the concept of planes in gt2 fss. The gt2 fcm is aim to the old xie beni cvi.”

FIGURE 3.13: Input is (introduction + conclusion) from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights are taken from <https://www.sciencedirect.com/science/article/pii/S0010482514001565>.

3.9 Comparison with Previous Works

TABLE 3.7: Comparison of the performance of the our three proposed model with that of other approaches for CSPubSum dataset. F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on different inputs from the CSPubSum dataset.

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore
LSTM Classification [43]	—	12.7	29.50	—	—	—
Gradient Boosting Regressor [33]	—	13.9	31.60	—	—	—
PGM + GloVe + Coverage [163]	31.46	8.57	29.14	12.01	15.54	85.31
PGM + ELMo + Coverage [162]	38.4	13.32	35.45	30.61	57.94	86.65
PGM + SciBERT + Coverage [164]	38.26	14.26	35.51	32.62	20.19	86.65

Table 3.7 compares the performance of our three proposed approach ($PGM + GloVe + Coverage$), ($PGM + ELMo + Coverage$) and ($PGM + SciBERT + Coverage$) with other competitive baselines in the literature, namely, an *LSTM-based* extractive summarization model [43], a *gradient boosting regression* extractive summarization model [33] on the CSPubSum dataset in terms of the F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, METEOR score, MoverScore and BERTScore metrics. ROUGE-2 F1-score and ROUGE-L F1-score of the LSTM-based model in [43] are 12.7 and 29.50, respectively while those in the gradient boosting regression model [33] are 13.9 and 31.6, respectively. All the above methods use extractive summarization on the full text (sans abstract) of the paper, that is, they select a set of sentences from a given document for inclusion in the research highlights. Our first work in this chapter’s thread [163] used an abstractive approach to generate research highlights from abstracts alone. The best-performing model in this work is a pointer-generator network with coverage and GloVe embeddings ($PGM + GloVe + Coverage$), achieving ROUGE-1 F1, ROUGE-2 F1, ROUGE-L F1, METEOR score, MoverScore F1 and BERTScore F1 scores of 31.46, 8.57, 29.14, 12.01, 15.54 and 85.31, respectively.

In a follow-up work, we proposed another method [162]: a pointer-generator network with coverage mechanism and ELMo embeddings($PGM + ELMo + Coverage$), which generally (i.e., in all metrics except ROUGE-2) outperforms the other above methods, namely, LSTM Classification, Gradient Boosting Regression, and ($PGM + GloVe + Coverage$).

In the last work in this chapter’s thread [164], we proposed a pointer-generator network with a coverage mechanism and SciBERT embeddings ($PGM + SciBERT + Coverage$), which achieves higher performance (except in MoverScore) than all the above models.

Note that here we have measured the performance on the holdout test set. Our method establishes a new state-of-the-art for the CSPubSum dataset. Note that our

model (PGM + SciBERT + Coverage) uses only abstracts as input, unlike the methods in [43] and [33], which use the full text of the paper. Since abstracts are much shorter than the main text, the computational overhead is significantly reduced. We emphasize that our methods simultaneously achieve higher performance as well as lower computational overhead.

3.10 Discussion

In this section, we discussed on the insights gained from our exploration of various models, including the pointer-generator model with GloVe embeddings and coverage mechanism [163], the pointer-generator model with ELMo embeddings and coverage mechanism [162], and the pointer-generator model with coverage mechanism and SciBERT Word embeddings [164]. We discuss potential directions for further development and research based on these findings.

1. The pointer-generator model incorporating GloVe embeddings and a coverage mechanism is designed to generate research highlights from a paper in an abstractive fashion. Unlike traditional methods that predominantly use extractive techniques, our model aims to create highlights that more naturally represent the paper’s content. However, there is still required for enhancement, particularly in refining the output to remove any irrelevant words or phrases. Further fine-tuning could also help the model better handle specific terminology, ensuring that the generated highlights are both concise and effectively summarize the paper.
2. The pointer-generator model, when combined with ELMo embeddings and a coverage mechanism, is more effective compared with other methods for generating research highlights. It has been observed that highlights generated with this model, using the abstract as input, tend to be more semantically meaningful than those generated from other input combinations. This approach is notably abstractive, unlike the primarily extractive methods used in current research. One of the key benefits of using the pointer-generator model with ELMo embeddings is the reduction in syntactic errors in the generated highlights, compared to other techniques. Despite these advancements, there is still a need for further refinement. Future research should aim to improve the coherence of the highlights and ensure they more accurately reflect the core contributions of the paper. Addressing these areas will be essential for enhancing the quality and effectiveness of the generated research highlights.
3. The pointer-generator model with SciBERT and a coverage mechanism achieved the best performance overall. We observed the best results when using the abstract

of a paper as input, rather than combining other sections. This method represents an abstractive approach to generating highlights from the paper. However, the generated highlights still exhibit limitations in syntax and semantics.

3.11 Summary

In this chapter, we proposed three abstractive models for generating research highlights from research papers. The significance of the proposed approach lies in its lightweight nature and wide applicability. Experimental results reveal that it is practically usable in major academic contexts. However, the approach can be further improved by integrating additional contextual information, such as syntax and semantics, which may not always be perfect this is a direction we plan to explore in future work. From our experiments, we observed that different word embeddings significantly improve performance across various metrics, including ROUGE, METEOR, MoverScore, and BERTScore. In this chapter, we focused only on highlight generation for computer science publications. Therefore, we plan to construct datasets for highlight generation in other scholarly domains, as well as datasets for very short summaries that are even easier to read than highlights.

4

A New Multidisciplinary Dataset for Generating Research Highlights

A robust and diverse dataset is crucial for effectively training a supervised model for research paper summarization or highlight generation. The dataset serves as the foundation upon which the model learns to extract key information and present it succinctly. High-quality datasets that cover a wide array of research topics, writing styles, and structures improve the model’s generalization ability, enabling it to perform well across different domains. Moreover, well-curated datasets help reduce biases and ensure that the generated summaries or highlights accurately reflect the original research’s intent and contributions. In the absence of such comprehensive datasets, the model’s output could be incomplete or misleading. Thus, the quality and diversity of the training data are crucial in determining the model’s effectiveness and reliability. This chapter introduces *MixSub* [164], a novel multidisciplinary dataset that includes research papers from diverse subject areas, each accompanied by author-written highlights.

4.1 Background and Motivation

This section provides an overview of the background related to research highlight generation and explains the motivation behind our proposed dataset.

In automated text summarization, generating brief and insightful research highlights

is essential for effectively communicating the main findings of scientific studies. These highlights allow researchers to grasp the key points of a paper swiftly, without having to read through the whole document. Nevertheless, the effectiveness of summarization models heavily relies on the presence of high-quality datasets that include pairs of research papers and their corresponding author-written research highlights.

The CSPubSum dataset, introduced by Collins et al. [43], includes URLs for approximately 10,000 computer science publications sourced from ScienceDirect¹. To our knowledge, it is the first benchmark dataset designed specifically for automatic highlight extraction. The dataset CSpubSum, which we crawled from ScienceDirect, contains full texts with abstracts and author-written research highlights. While this dataset has made a notable impact in the field of research highlight generation from scientific papers, its application is confined to computer science papers, limiting the generalizability of models trained on it to other scientific disciplines.

To overcome this limitation, Cagliero et al. [33] released two additional datasets, AIPubSumm and BioPubSumm, to extend the range of domains. AIPubSumm features 198 training articles and 66 test articles focused on Artificial Intelligence, whereas BioPubSumm includes 8,070 training articles and 2,690 test articles from the Biomedical field. Both datasets come with URLs sourced from ScienceDirect. Despite these efforts, both datasets remain somewhat limited in scope, size, and specific subject domains.

To address these limitations, we present a new multi-disciplinary dataset called *MixSub* [164]. *MixSub* consists of research papers along with author-written highlights from different subject domains. The goal of this dataset is to offer a wider and more diverse selection of research topics, thereby strengthening the effectiveness and adaptability of summarization models. By providing a comprehensive and varied dataset, *MixSub* aims to enhance the performance and relevance of summarization systems, benefiting the broader research community.

4.2 Challenges and Opportunities

This section addresses the challenges associated with current datasets used for generating research highlights and explores the opportunities provided by the newly introduced dataset.

4.2.1 Challenges

1. **Domain Specificity and Generalization:** Existing datasets for research highlights generation often focus on specific fields like computer science or biomedicine.

¹<https://www.sciencedirect.com/>

This focus can limit the effectiveness of summarization models in other areas. Models trained on these specialized datasets may struggle to adapt to different domains, affecting their overall flexibility. Creating datasets that cover a variety of scientific fields is challenging due to differences in terminology, writing styles, and content structures across disciplines.

2. **Dataset Size and Quality:** Creating large datasets while ensuring they remain high quality is difficult. Achieving consistent and accurate annotations, along with handling the costs and resources needed for data collection and annotation, presents a major challenge.

4.2.2 Opportunities

1. **Expanding Domain Coverage:** Creating multidisciplinary datasets offers an opportunity to develop more adaptable models for research highlight generation that work well across different research areas. This approach enhances the performance of research highlight systems by incorporating a wide range of scientific research papers.

4.3 Dataset Construction

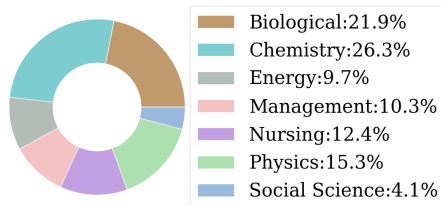
This section details the creation of the *MixSub dataset* [164], including its data collection, curation, and organization into training, validation, and test subsets. We also provide a comparative overview with the *CSPubSum* dataset, highlighting key statistics and characteristics.

We proposed a new dataset called *MixSub* that contains research articles from multiple domains. To prepare *MixSub*, we crawled the ScienceDirect website and curated articles published in various journals in year 2020. We removed the articles that did not contain research highlights. Finally, we got 19785 articles with author-written research highlights as shown in Table 4.1. Each example in this dataset is organized as (*abstract*, *author-written research highlights*). We have also segmented the dataset into training, validation and test subsets. In this corpus, the average abstract size is 148 words while that of highlights is 57. For 72% of the papers, highlights are 1.5 times or more shorter than the abstract. We split each category of documents into training: validation: test subsets in the ratio 80 : 10 : 10. We have grouped similar journal papers according to their domain as shown in Table 4.1 and also highlighted using a pie chart Figure 4.1. A summary of the above two datasets is shown in Table 4.2. Table 4.2 presents a comparative summary of our contributed dataset, *MixSub*, alongside the *CSPubSum* dataset. The table highlights key statistics for both datasets, including training, validation, and

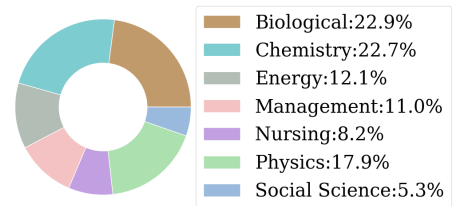
test sets, average word counts for abstracts and highlights, and the percentage of article pairs with significant compression.

TABLE 4.1: Subject-wise URL count in MixSub dataset.

Domain Name	Subject name	#Total	#Train	#Val	#Test
Biological	Agricultural and Biological Sciences	2156	1726	216	214
	Biochemistry, Genetics and Molecular Biology	976	806	71	99
	Immunology and Microbiology	233	195	24	14
	Neuroscience	962	771	96	95
Chemistry	Chemical Engineering	2140	1713	214	213
	Chemistry	2282	1919	240	123
	Materials Science	735	572	82	81
Energy	Energy	1313	1025	145	143
	Environmental Science	677	517	81	79
Management	Business, Management and Accounting	698	560	70	68
	Decision Sciences	947	759	95	93
	Economics, Econometrics and Finance	421	324	56	41
Nursing	Health Sciences	823	796	12	15
	Nursing and Health Professions	61	47	8	6
	Pharmacology, Toxicology, Pharmaceutical Science	1184	949	118	117
	Psychology	28	24	3	1
	Veterinary Science and Veterinary Medicine	186	156	19	11
Physics	Earth and Planetary Sciences	1354	1038	159	157
	Mathematics	288	232	29	27
	Physics and Astronomy	1469	1177	147	145
Social Science	Social Sciences	852	654	100	98



(a) Distribution of train examples.



(b) Distribution of test examples.

FIGURE 4.1: Subject-wise distribution of papers in MixSub dataset.

TABLE 4.2: Comparison of Dataset Statistics: CSPubSum vs. MixSub.

Dataset	Train	Val	Test	Average Words (abstract)	Average Words (highlight)	% of article-pairs where compression ≥ 1.5 times
CSPubSum [43]	8115	1014	1013	186	52	98
MixSub (ours)	15960	1985	1840	148	57	72

4.4 Experimental Setup

In this section, we discussed the datasets used, the data pre-processing steps, and the implementation details.

4.4.1 Datasets

We used our contributed dataset *MixSub*, which includes URLs for approximately 19K in the field of research articles from multiple domains from ScienceDirect. We have crawled the dataset. Every document contains the following fields: abstract, research highlights written by the authors found in a research paper. Each example in the dataset is organized as (abstract, author-written research highlights). The dataset was divided into training, validation, and test subsets, following an 80 : 10 : 10 ratio.

4.4.2 Data Pre-Processing

We have used the Stanford CoreNLP Tokenizer² for tokenizing the sentences. The whole corpus is first converted to lowercase. We have removed all unnecessary symbols, letters, and other elements from the text that do not affect the aim of our research. In particular, HTML tags, parentheses, and special characters have been removed. We have rearranged our dataset to include (*abstract*, *author-written research highlights*). In the future, we might explore using full text or sections of the full text for MixSub as well. When the abstract is used as the input, the maximum number of input tokens is limited to 400, while the generated research highlights are restricted to 100 tokens for all models.

4.4.3 Implementation Details

We trained four variants of the proposed model, as described in Chapter 3, Section 3.8: (1) a pointer-generator network with word embeddings trained from scratch during model training (**PGM**), (2) a pointer-generator network with a coverage mechanism, where word embeddings are also trained from scratch (**PGM** + **Coverage**), (3) a

²<https://stanfordnlp.github.io/CoreNLP/>

pointer-generator network with SciBERT embeddings for the input tokens (**PGM + SciBERT**), and (4) a pointer-generator network combining both the coverage mechanism and SciBERT embeddings for the input tokens (**PGM + SciBERT + Coverage**). For all SciBERT model variants, the embeddings are fine-tuned during training. We trained all models on Tesla P100-PCIE-16GB Colab Pro+ that supports GPU-based training. We used mini-batches of size 16. For all models, we used bidirectional LSTMs with cell size of 256. For models without SciBERT, word embeddings of dimension 128 are trained end-to-end with the model. For models with SciBERT, pre-trained word embeddings of dimension 768 are used. For all experiments, we constrained the vocabulary size to the most frequent 50,000 tokens. We considered gradient clipping with a maximum gradient norm of 1.2. Out of the four variations of the SciBERT model, we use `SciVocab-uncased`³. We used other hyperparameters as suggested by [174]. We have used the validation set to determine the number of epochs for training.

4.5 Evaluation and Results on the MixSub Dataset

In this section, we report the results of experiments on the *MixSub* dataset. We trained the models in two ways:

- **Case 1:** We trained all the four models on each subject cluster separately and tested them on the corresponding test documents.
- **Case 2:** We did not distinguish between the subject categories of the papers but simply collected all the documents of the training corpus, and trained the models. Then we evaluated them on the test corpus and reported the results for each subject category.

Note that in each case, the input is only the abstract of a paper. Since MixSub currently does not contain the body of a paper, we cannot use other sections of a paper as the input. Results are reported in Table 4.3 for F1-score of ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore and BERTScore. The top row labeled ‘Full MixSub’ shows the results when the models are trained on the whole training corpus without regard to the specific subject category of the papers and tested on the test corpus, again without regard to the specific subject category of the papers. The remaining rows show the scores obtained on each category of papers when the models are trained either on the respective clusters (Case 1) or on the whole training corpus without regard to subject category (Case 2). We observe that among the four models, (**PGM + SciBERT + Coverage**) achieves the highest ROUGE, METEOR, MoverScore, and BERTScore

³https://huggingface.co/allenai/scibert_scivocab_uncased/

values. We observed that sometimes training on subject-specific clusters leads to higher scores and at other times, training on the whole corpus produces better scores at the subject level. (**PGM** + **SciBERT** + **Coverage**) consistently outperforms all the other models in all cases.

TABLE 4.3: Evaluation of pointer-generator type models with and without SciBERT embeddings: F1-scores for ROUGE, METEOR, MoverScore and BERTScore on MixSub dataset. The first row (where dataset is ‘Full MixSub’) indicates the performance when the models are trained on the whole MixSub training set and evaluated on the whole MixSub test set, without distinguishing between the subject categories of the papers. In the remaining part of the table, two cases are considered: Case 1: Trained on each subject-cluster of MixSub training set and evaluated on the corresponding test set; Case 2: Trained on the entire MixSub training set and evaluated on each subject-cluster of MixSub test set. All scores are presented as percentages (%).

Dataset	Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore
Full MixSub	abstract only	PGM	29.3	8.43	26.99	21.46	13.38	83.41
		PGM + Coverage	31.52	9.18	29.21	22.91	15.80	85.22
		PGM + SciBERT	30.44	9.68	27.81	23.38	13.86	84.83
		PGM + SciBERT + Coverage	31.78	9.76	29.3	24	15.85	85.25
Biological	Case 1	PGM	25.4	5.13	23.56	18.02	11.21	83.51
	Case 2	PGM	27.88	7.36	25.77	9.29	12.49	81.01
	Case 1	PGM + Coverage	28.23	6.18	25.96	19.60	11.79	83.99
	Case 2	PGM + Coverage	28.76	7.74	26.76	9.89	13.02	82.01
	Case 1	PGM + SciBERT	28.74	7.45	26.56	20.87	12.10	84.49
	Case 2	PGM + SciBERT	28.42	8.01	26.02	9.92	12.34	81.1
	Case 1	PGM + SciBERT + Coverage	29.9	7.6	27.57	21.53	13.50	84.74
	Case 2	PGM + SciBERT + Coverage	28.88	8.03	26.76	9.98	13.43	84.72
Chemistry	Case 1	PGM	27.83	7.39	26.1	13.58	13.72	82.99
	Case 2	PGM	27.44	7.15	25.55	9.63	13.63	81.83
	Case 1	PGM + Coverage	29.67	8.09	27.4	14.14	14.5	83.02
	Case 2	PGM + Coverage	29.68	7.73	27.57	9.88	16.26	82.11
	Case 1	PGM + SciBERT	30	8.58	27.98	22.38	14.5	84.93
	Case 2	PGM + SciBERT	29.16	8.47	26.96	9.85	14.9	81.92
	Case 1	PGM + SciBERT + Coverage	31.4	8.9	29.33	24.37	16.84	85.11
	Case 2	PGM + SciBERT + Coverage	30.41	8.58	28.15	10.19	16.61	82.19
Energy	Case 1	PGM	23.81	4.21	21.99	15.80	8.78	83.44
	Case 2	PGM	29.33	8.56	26.91	9.40	13.65	81.07
	Case 1	PGM + Coverage	27.12	4.71	24.98	18.11	11.75	84.05
	Case 2	PGM + Coverage	31.69	9.24	29.25	10.42	16.0	82.55
	Case 1	PGM + SciBERT	28.87	6.18	26.5	19.92	13.80	84.53
	Case 2	PGM + SciBERT	29.61	9.09	26.62	9.70	13.22	81.9
	Case 1	PGM + SciBERT + Coverage	30.04	6.84	27.4	20.85	16.45	85.51
	Case 2	PGM + SciBERT + Coverage	32.15	9.66	29.7	10.77	16.24	82.84
Management	Case 1	PGM	32.39	8.73	30.08	15.80	14.3	83.18
	Case 2	PGM	34.51	11.68	31.64	9.89	15.5	81.53
	Case 1	PGM + Coverage	34.47	9.54	31.77	18.23	17.6	83.67
	Case 2	PGM + Coverage	37.25	13.23	34.4	10.54	18.94	82.19
	Case 1	PGM + SciBERT	33.54	9.15	30.9	23.17	15.5	85.20
	Case 2	PGM + SciBERT	35.65	13.18	32.81	10.66	16.33	82.26
	Case 1	PGM + SciBERT + Coverage	36.05	11.02	33.27	25.24	17.9	85.66
	Case 2	PGM + SciBERT + Coverage	38.39	13.62	35.64	11.39	18.96	83.03
Nursing	Case 1	PGM	25.2	4.82	22.95	17.01	8.3	83.18
	Case 2	PGM	28.64	7.9	26.08	8.55	11.76	81.08
	Case 1	PGM + Coverage	28.2	5.46	25.5	18.79	10.85	83.71
	Case 2	PGM + Coverage	30.38	8.39	27.83	9.78	14.11	81.61
	Case 1	PGM + SciBERT	30.21	6.71	27.54	21.16	12.3	84.21
	Case 2	PGM + SciBERT	31.43	9.43	28.42	9.52	13.92	81.04
	Case 1	PGM + SciBERT + Coverage	31.61	8.09	28.7	22.73	14.93	84.57
	Case 2	PGM + SciBERT + Coverage	31.61	10.28	28.83	9.98	15.18	81.42
Physics	Case 1	PGM	29.97	7.98	27.44	21.19	12.92	84.19
	Case 2	PGM	30.41	9.07	28.07	10.21	13.97	81.05
	Case 1	PGM + Coverage	31.06	7.9	28.52	21.26	14.52	84.95
	Case 2	PGM + Coverage	32.05	10.26	30.27	10.51	16.11	82.4
	Case 1	PGM + SciBERT	30.67	8.3	28.45	21.76	13.4	85.02
	Case 2	PGM + SciBERT	31.31	10.67	28.6	10.38	13.99	81.93
	Case 1	PGM + SciBERT + Coverage	32.13	8.92	29.53	22.83	16.6	85.25
	Case 2	PGM + SciBERT + Coverage	32.99	11.01	30.35	11.16	16.76	82.45
Social Science	Case 1	PGM	22.64	4.36	20.51	13.94	12.4	81.86
	Case 2	PGM	30.23	10.39	27.29	9.23	11.79	81.64
	Case 1	PGM + Coverage	26.96	5.17	24.19	16.08	13.85	82.72
	Case 2	PGM + Coverage	31.99	11.2	28.82	9.61	13.77	81.69
	Case 1	PGM + SciBERT	30.11	8	26.87	19.02	12.68	83.35
	Case 2	PGM + SciBERT	31.21	10.97	27.75	9.63	11.83	81.65
	Case 1	PGM + SciBERT + Coverage	31.89	8.67	28.36	19.99	13.95	83.81
	Case 2	PGM + SciBERT + Coverage	32.35	11.8	28.87	9.79	13.73	81.75

4.6 Case Studies on MixSub Dataset

We now present a few examples demonstrating the outputs generated by the pointer-generator type models used in this paper. We now present an example demonstrating the output produced by the four variants of pointer-generator model for the *MixSub* dataset. Figure 4.2 displays the outputs when the models take only the abstract as the input. We observe that the last two sentences produced by the plain pointer-generator model are identical. This issue goes away when coverage is added but now the model’s output is not entirely factually correct: it generates “The reflection of the plasma photonic crystal to near infrared radiation *increases* with the wave angle.” while the abstract mentions “The reflection of the plasma photonic crystal to near infrared radiation *decreases* with increasing of the incident wave angle ...”. This factual error disappears when SciBERT is added. The output quality improves further when coverage is added. While PGM with SciBERT (but without coverage) outputs “the near infrared filter photonic crystal is adjusted”, the final model (PGM + SciBERT + Coverage) is more precise and meaningful: “*infrared radiation pass band* can be adjusted”. Therefore, the highlights produced by (PGM + SciBERT + Coverage) seem to be most satisfactory though its last sentence has a syntax error due to a missing relative pronoun: “A transfer method is proposed for infrared radiation pass band [which] can be adjusted ...”.

Author-written research highlights: “A plasma photonic crystal composed of ito and plasma is proposed. The performance of ppc in near infrared radiation modulation is researched by tmm. The near infrared radiation pass band can be adjusted by plasma frequency. The ppc has a potential application in tunable near infrared filter devices.”
PGM: “A plasma photonic crystal for infrared radiation modulation was proposed. The plasma photonic crystal was researched by the changing changing of plasma frequency of plasma. The incident wave angles have little effect on the transmission of plasma. The incident wave angles have little effect on the transmission of plasma.”
PGM + Coverage: “A plasma photonic crystal for infrared radiation and plasma is proposed. The incidence wave angles can be adjusted by the changing plasma oxide band. The incident wave angles have little effect on the transmission of plasma photonic crystal. The reflection of the plasma photonic crystal to near infrared radiation increases with the wave angle.”
PGM + SciBERT: “A plasma photonic crystal for infrared radiation modulation is proposed. The near infrared filter photonic crystal is adjusted by the changing of plasma frequency of plasma photonic crystal in near infrared filter devices. The proposed plasma photonic crystal has a potential application in tunable near infrared filter devices.”
PGM + Coverage + SciBERT: “A plasma photonic crystal for infrared radiation modulation is proposed. A transfer matrix method is proposed for infrared radiation pass band can be adjusted by the changing of plasma frequency. The proposed plasma photonic crystal has a potential application in tunable near infrared filter devices.”

FIGURE 4.2: Input is only the abstract of an article from the MixSub dataset. Highlights produced by the pointer-generator type models with and without SciBERT embeddings are shown. Input and author-written research highlights taken from <https://www.sciencedirect.com/science/article/pii/S1567173920301292>.

4.7 Discussion

In this chapter, we leveraged the *MixSub* dataset to evaluate various summarization models. The (*PGM + SciBERT + Coverage*) model consistently outperformed others in ROUGE, METEOR, MoverScore and BERTScore metrics, demonstrating the effectiveness of integrating a coverage mechanism with SciBERT embeddings to enhance summary quality.

The *MixSub* dataset, which includes research papers and author-written highlights from multiple domains, was crucial for a comprehensive evaluation. It allowed us to test and compare different summarization techniques effectively. We observed that training on subject-specific clusters sometimes resulted in better performance, suggesting that specialized data can improve summary relevance for particular topics. However, training

on the entire corpus also proved beneficial, indicating that broader context can enhance summary accuracy.

These insights underscore the importance of the *MixSub* dataset in providing a detailed assessment of summarization techniques and highlight the variability in model performance based on training strategies.

4.8 Summary

In conclusion, the *MixSub* dataset has significantly advanced text summarization techniques. The (*PGM* + *SciBERT* + *Coverage*) model achieved the highest scores across key metrics, demonstrating the effectiveness of combining advanced summarization techniques.

The dataset’s extensive coverage of research papers and highlights across various subjects provided a robust foundation for evaluating model performance and understanding the impact of different training approaches. The *MixSub* dataset has proven invaluable for assessing summarization models and making significant contributions to the field through detailed analysis and comparisons.

Future research should build on these findings by exploring additional models and datasets and incorporating new techniques to further enhance the quality of the generated research highlights.

5

Entity-Driven Insights: Named Entity Recognition-Based Automatic Generation of Research Highlights

This chapter investigates the use of deep learning techniques for generating research highlights, with a specific emphasis on named entity recognition (NER). In this work [165], we propose a mechanism that integrates named entity recognition with a deep learning model to automatically generate research highlights from scientific documents. This approach leverages named entity recognition (NER) to improve the precision, relevance, and meaningfulness of the generated summaries, making a significant contribution to the automation of research highlight generation.

5.1 Introduction

The rapid increase in scientific publications, with the volume doubling roughly every nine years [27, 194], presents a significant challenge for researchers trying to stay up-to-date in their fields. To address this, publishers often require authors to include succinct research highlights alongside their abstracts, providing a quick overview of the paper’s core contributions. Automatic text summarization aims to distill documents by capturing essential information from the original text. However, scientific papers, being

longer and more complex than news articles, pose unique summarization challenges.

As the volume of scientific literature continues to surge, the demand for effective summarization tools grows more pressing. Traditional summarization methods [53, 210] often struggle with scientific texts, particularly with accurately representing **multi-word named entities**.

Named Entity Recognition (NER) is a crucial technique in text analysis that identifies and classifies specific entities such as name's of people, locations, and organizations. By correctly grouping these entities, NER helps maintain the coherence and relevance of terms within a document, which is essential for effective information extraction and summarization. For instance, [72] proposed incorporating appropriate weights for named entity tags in the SweSum summarizer, specifically for Swedish newspaper texts. Additionally, [116] developed an extractive summarization technique that evaluates sentence significance based on the density of named entities.

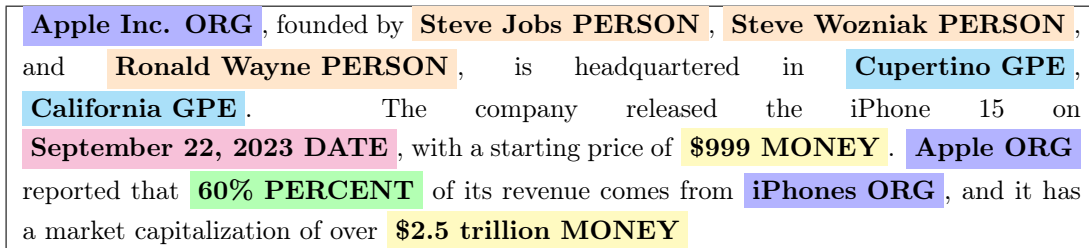
In scientific research papers, NER becomes even more complex due to the varied patterns of named entities across different domains. Handling **multi-word named entities** accurately is a significant challenges, as these entities often consist of specialized and intricate terms that can be misrepresented if not processed correctly.

This chapter introduces a novel approach to addressing this challenge by integrating named entity recognition(NER) with the abstractive summarization techniques. Our approach treats **multi-word named entities** as single units during the summarization process or highlights generation process, aiming to prevent common issues such as entity fragmentation and misrepresentation. This method seeks to enhance the precision and clarity of generated research highlights.

By improving the handling of domain-specific entity patterns, our method aims to offer a more reliable and coherent generation of research highlights, providing a more effective solution for summarizing complex scientific documents. This chapter explores how this integration can lead to more accurate and insightful summaries, ultimately assisting researchers in navigating and understanding the expanding body of scientific knowledge and enhancing automatic summarization systems.

5.2 Background and Motivation

This section provides an overview of the background related to research highlight generation and the motivation behind this research.



Apple Inc. **ORG**, founded by **Steve Jobs PERSON**, **Steve Wozniak PERSON**, and **Ronald Wayne PERSON**, is headquartered in **Cupertino GPE**, **California GPE**. The company released the iPhone 15 on **September 22, 2023 DATE**, with a starting price of **\$999 MONEY**. **Apple ORG** reported that **60% PERCENT** of its revenue comes from **iPhones ORG**, and it has a market capitalization of over **\$2.5 trillion MONEY**.

FIGURE 5.1: Named entity recognition example, showcasing entities like organizations, people, locations, dates, monetary values, and percentages.

5.2.1 Background

Named entity recognition (NER) involves identifying and categorizing distinct elements in a text that exhibit common traits, such as names of people, places, organizations, as well as fixed categories like time, dates, percentages, monetary values, and more, as shown in Figure 5.1. The concept of named entities became significant with the introduction of the Message Understanding Conference [66], where identifying these entities in text emerged as a crucial task. NER plays a vital role in various *information extraction* processes, allowing systems to identify and categorize important elements within a document.

In the biomedical field, biomedical named entity recognition (NER) focuses on detecting and classifying specialized terms in molecular biology, with the goal of identifying key concepts relevant to medicine and bio-science. These concepts can include genes, proteins, diseases, drugs, body parts, tissues, and specific locations of activity, such as organisms or cells [184]. In chemistry, chemistry named entity recognition (ChemNER) [46] has evolved from focusing on drug-related entities to covering a broader spectrum of elements, including chemicals, chemical-disease interactions, and drug-chemical-protein relationships. BioNER [184] complements this advancement by employing text mining to uncover essential interactions between drugs, chemicals, proteins, and chemical-disease connections. In the computer science domain, CsNER [51] encompasses entities like research problems, methods, solutions, techniques, resources, evaluations, and tools. Evaluations focus on corpus size (including tokens, papers, and entities), domain coverage, semantic types, annotation details, and annotation methods [147, 51]. Common tools used for Named Entity Recognition (NER) include NLTK ¹, SpaCy ², Stanford NER ³, and Hugging Face Transformers ⁴. But in our work, instead of finding or labeling entity names from scientific publications as mentioned above, we group **multi-word**

¹<https://github.com/nltk/nltk>

²<https://spacy.io/api/entityrecognizer>

³<https://nlp.stanford.edu/software/CRF-NER.shtml>

⁴<https://huggingface.co/>

named entities together as **single tokens** rather than breaking them into multiple tokens. For example, instead of treating the individual tokens “deep”, “convolutional”, “neural”, and “networks” separately, we combine them into a single token, “deep convolutional neural networks” (referenced as vocab index 1). Similarly, instead of treating the individual tokens “support”, “vector”, and “machines”, we pass all the three tokens together as a single token “support vector machines” (referenced as vocab index 14).

5.2.2 Motivation

In the context of scientific text summarization and the generation of research highlights, it is crucial not only to identify named entities but also to group similar entities together rather than treating them as individual tokens. This ensures that the summarization process retains the coherence and relevance of related terms, which is essential for producing accurate and meaningful summaries.

To address this, we explored the use of deep learning techniques in combination with Named Entity Recognition (NER) to enhance the quality of generated research highlights. By integrating NER into the summarization process, we aim to improve the quality of the generated highlights, ensuring they are more accurate, contextually relevant, and effective in capturing the key points of the scientific papers.

5.3 Challenges and Opportunities

This section discusses the current challenges in generating research highlights and correctly generating named entities and explores potential advancements in the field through innovative methods and technologies.

5.3.1 Challenges

1. **Handling Fragmentation of Named Entities:** One significant challenge in abstractive summarization is accurately representing **multi-word named entities**. Deep learning models may sometimes split or misinterpret these entities, which can lead to distortions in meaning and coherence in the summaries. Ensuring that such entities are correctly handled is crucial for maintaining the semantic integrity of the generated content.
2. **Complex Integration of NER with Summarization Models:** Integrating named entity recognition (NER) with abstractive summarization models adds complexity to the pre-processing and model interaction. The challenge lies in ensuring that NER outputs are seamlessly incorporated into the summarization process without introducing errors or negatively impacting overall performance.

3. **Domain-Specific Generalization:** Named entities and their significance can vary significantly across different scientific domains. Creating a system that effectively generalizes across diverse fields while maintaining high accuracy in entity recognition and summarization is a challenging task.
4. **Maintaining Text Coherence:** While treating named entities as single tokens can prevent fragmentation, it may also compromise the fluency and coherence of the generated text. Balancing the preservation of entity integrity with the production of natural, readable summaries presents a significant challenge.

5.3.2 Opportunities

1. **Improved Summarization Accuracy:** Integrating NER into the summarization process can enhance the accuracy of the generated highlights. By ensuring that named entities are treated correctly, the summaries can become more precise and closely aligned with the original content.
2. **Enhanced Semantic Consistency:** Using NER to handle named entities as single units helps maintain semantic consistency, which is vital for producing summaries that accurately reflect the key information and context of the original research papers.
3. The integration of named entity recognition (NER) models with deep learning techniques opens new possibilities for generating meaningful and accurate research highlights. Exploring innovative methods for handling entities can drive advancements in research highlights generation tasks or automatic text summarization systems.

5.4 Main Contributions

The main contributions of this chapter are:

1. We propose a mechanism to combine named entity recognition with pointer-generator networks having coverage mechanism to automatically generate research highlights, given the abstract of a research paper. To the best of our knowledge, this work is the first attempt to use NER in pointer-generators with coverage mechanism [174] to generate research highlights.
2. We analyze the performance of generating research highlights for the following different input types: (a) the input is the abstract only, (b) the input comprises

the abstract and the conclusion, (c) the input comprises the introduction and the conclusion.

3. We evaluate the performance of the models using ROUGE [102], METEOR [17], MoverScore [227], and BERTScore [224] metrics.

5.5 Methodology

In this section, we have described the model we use to generate research highlights from the scientific papers.

We used a pointer-generator network [174] as our baseline model. While the pointer-generator model [174] first tokenizes a document using Stanford CoreNLP tokenizer and converts the tokens to word embeddings (trained with the model), the method we propose here performs NER on the input document and considers a multi-word named entity as a single token when training the model. We perform experiments with 4 variants: (1) the original pointer-generator model proposed in [174] (**PGM**), (2) pointer-generator model integrating coverage mechanism (proposed in [191]) (**PGM + Coverage**), described in the same work [174], (3) NER-based pointer-generator model (**NER + PGM**), and (4) NER-based pointer-generator model with coverage mechanism (**NER + PGM + Coverage**). The architecture of our model is shown in Figure 5.2.

5.5.1 NER-based Pointer-Generator Network

This model consists of an NER-based tokenizer layer and a pointer-generator network. The NER-based tokenizer layer converts the words in the input document to a sequence of tokens, thus preserving an entity name as a single token. In particular, it uses the named entity recognizer in spaCy⁵ However, we do not use entity types. We do not use pre-trained word embeddings as [129] do; in our case token embeddings are learned from scratch during training. Here, the main role of NER is that instead of directly feeding the normal tokens of the input document into the encoder, we are passing the NER-based tokens. Later, we added a coverage mechanism [191] to address the issue of repetitive phrase generation, which helps to improve the diversity and relevance of the generated phrases.

⁵<https://spacy.io/usage/linguistic-features>.

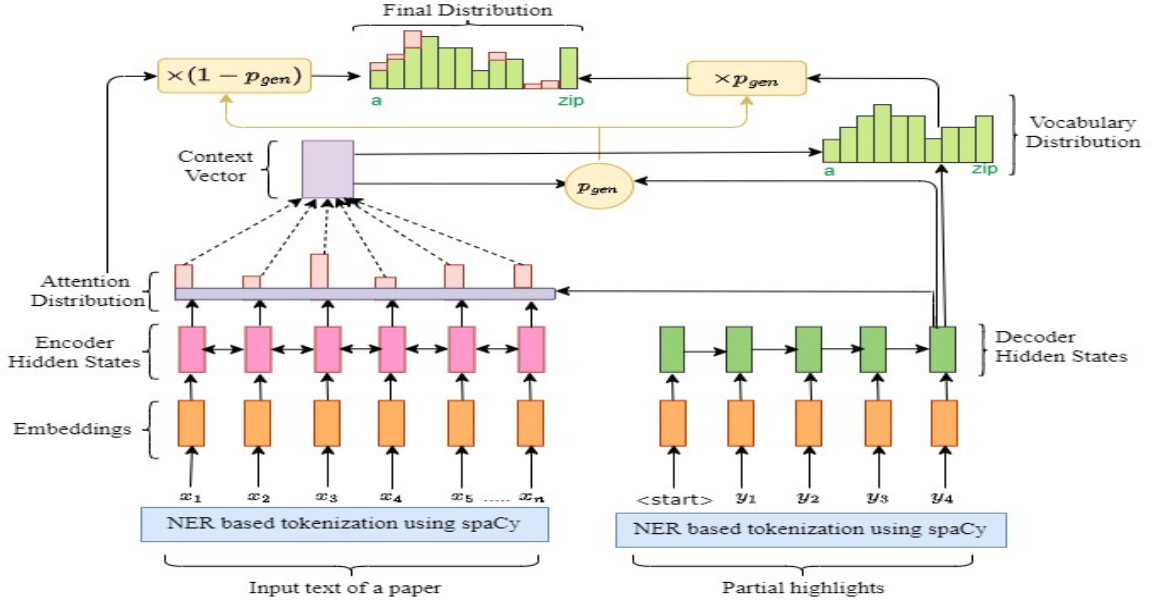


FIGURE 5.2: Proposed model: NER-based pointer-generator network with coverage mechanism.

5.6 Experimental setup

In this section, we discussed the datasets used, the data processing steps, and the implementation details.

5.6.1 Dataset Details

We use the data sets published by Collins et al. [43], which contains URLs of 10147 computer science publications from ScienceDirect (<https://www.sciencedirect.com/>). Title, abstract, author-written research highlights, a list of keywords referenced by the authors, introduction, related work, experiment, conclusion, and other important sub-sections as found in typical research papers are all included for each document. In our setup, every example in this data set is organised as follows: (*abstract, author-written research highlights, introduction, and conclusion*). We use 8116 examples for training, 1017 examples for validation, and 1014 examples for testing.

5.6.2 Data Processing

We have removed digits, punctuation, and special characters from the documents and lowercased the entire corpus. The `retokenizer.merge` method of spaCy is used to tokenize and merge several tokens into one single token based on the named entities in the document. Instead of individual tokens of “artificial”, “neural”, and “network”, we pass

all the three tokens together as a single token “artificial neural network” (referenced as vocab index 17). The dataset is then reorganized in several ways to conduct various experiments. We organize the dataset as (*abstract, author-written research highlights*), (*abstract \oplus conclusion, author-written research highlights*), and (*introduction \oplus conclusion, author-written research highlights*) where ‘ \oplus ’ denotes text concatenation. Since abstract and introduction usually emphasize the same aspects of the paper, we have not included them together.

5.6.3 Implementation Details

In this CSPubSum dataset, the average abstract length is 186 tokens, while the average author-written research highlights length is 52 tokens. When we considered the abstract and conclusion as input, the average length was 643 tokens. When we considered the introduction and conclusion as input, the average length was 1234 tokens. Therefore, in our model, we have set the maximum number of input tokens to 400 when the abstract is used as input. In all other cases, the maximum allowed input token count is set to 1500. For all cases, the generated research highlights are limited to a maximum token count of 100. We trained all models on Tesla V100-SXM2-16GB Colab Pro+ that supports GPU-based training. We used mini-batches of size 16. For all models, we used two bidirectional LSTMs with cell size of 256, word embeddings of dimension 128, and maximum vocabulary size of 50K tokens. We considered gradient clipping with a maximum gradient norm of 1.2. We use other hyperparameters as suggested by [174].

5.7 Results

In this section, we present a detailed analysis of the results from our proposed model, using both automatic evaluation metrics and manual evaluation.

5.7.1 Comparison of Pointer-Generator type Models

In this sub-section, we report the results of experiments on the CSPubSum dataset for various input types. Table 5.1 shows the F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore, and BERTScore metrics for various inputs from the test dataset. Among the four models, the NER-based pointer-generator network with coverage mechanism (*NER + PGM + Coverage*) achieves the highest ROUGE, METEOR, MoverScore and BERTScore values, in all cases. It appears that treating an entity as a single token in the input helps to learn better embeddings and results in more controlled generation of the output, thereby reducing semantically invalid words and phrases. We aimed to investigate this aspect in future. The (*NER + PGM + Coverage*) model

achieves the highest scores when the input is the abstract, indicating that most of the findings reported by the research highlights are already in the abstract, and adding additional sections to the input contributes to noise for the model.

TABLE 5.1: Evaluation of pointer-generator type models with and without NER: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on various inputs from CSPubSum dataset. All our ROUGE scores have a 95% confidence interval of at most ± 0.25 as reported by the official ROUGE script.

Input	Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore
abstract only	PGM	35.44	11.57	29.88	25.4	18.47	83.80
	PGM + Coverage	36.57	12.3	30.69	25.4	19.67	84.05
	NER + PGM	35.88	12.78	33.21	29.14	18.57	86.02
	NER + PGM + Coverage	38.13	13.68	35.11	31.03	20.77	86.3
abstract + conclusion	PGM	29.85	8.16	25.80	19.38	14.18	83.19
	PGM + Coverage	31.70	8.31	26.72	20.92	15.73	83.49
	NER + PGM	35.12	12.37	32.37	28.34	17.64	86.08
	NER + PGM + Coverage	37.48	13.26	34.95	28.97	20.84	86.64
introduction + conclusion	PGM	29.78	7.47	25.15	19.25	14.23	83.05
	PGM + Coverage	31.63	7.65	26.25	20.24	15.76	83.32
	NER + PGM	31.74	9.18	29.44	23.82	15.11	85.78
	NER + PGM + Coverage	34.24	9.82	31.92	25.36	18.35	86.1

5.7.2 Manual Evaluation

We selected a set of 25 papers, their author-written highlights (**A**) and their highlights from only the ($NER + PGM + Coverage$) model (**M**). We recruited 16 human annotators (possessing or pursuing advanced degrees in software engineering at premier universities in India) to independently rate a given summary on a scale of 1(low) to 5(high) for adequacy and fluency (separately). Each rater was given the full text of a paper and *either* the author-written *or* the machine-generated highlights of the paper, but not told which one. Each summary was rated independently by two raters. On fluency, the average score for **A** was 4.02 and that for **M** was 3.3, while on adequacy, the average score for **A** was 3.82 and that for **M** was 3.12. This shows the machine-generated highlights are only slightly worse than the human-written ones.

5.8 Case Studies

Figure 5.3, 5.4 and 5.5 shows sample outputs generated by our models for various input types. In all the case studies reported below, we highlight examples of **factual errors**, **repeating words** and **correct named entities**. Note that the NER-based models correctly generate the named entities in the output.

In Figure 5.3, the pointer-generator model (PGM) incorrectly generates “2 Korea 2 image acquisition [UNK]”, “segment square lengths”, and “collection relaxation”. In

contrast, the pointer-generator model with the coverage mechanism ($PGM + Coverage$) generates these outputs correctly, addressing all errors made by the PGM model. However, it repeats a whole sentence "... approach based on the Lagrangian relaxation and ...". After integrating NER with the PGM model ($NER + PGM$), it generates a meaningful summary and correctly identifies the multi-word entity "Korean satellite". However, it repeats part of the phrase "Lagrangian relaxation and subgradient methods". With the addition of the coverage mechanism to the ($NER + PGM$) model, this issue of repeated phrases is resolved. Similarly, Figures 5.4 and 5.5 show that the summaries generated by the ($NER + PGM + Coverage$) model are more meaningful than those from other models

Author-written research highlights:	"This paper studies an image collection planning problem for a korean satellite kompsat 2(korea multi purpose satellite 2). We present a binary integer programming model for this problem in a multi orbit long term planning environment. A heuristic solution approach based on the lagrangian relaxation and subgradient methods is provided."
PGM:	"Image collection with 2 korea 2 image acquisition [UNK] of segment square lengths as well as well as collection relaxation . Integer programming model exploiting lagrangian relaxation. Results of our computational experiment based on the lagrangian relaxation and subgradient methods".
PGM + Coverage:	"Image collection planning problem with a multi orbit multi purpose satellite 2 korea multi purpose. A binary integer programming model for a multi orbit long term planning. A heuristic solution approach based on the lagrangian relaxation and subgradient methods. A heuristic solution approach based on the lagrangian relaxation and subgradient methods ."
NER + PGM:	"We present a binary integer programming model for this image collection planning problem for a korean satellite kompsat. We present a binary integer programming model for this problem in a multi orbit long term planning environment. Combining the lagrangian relaxation and subgradient methods using the lagrangian relaxation and subgradient methods to solve the top problem."
NER + PGM + Coverage:	"We present an image collection planning problem for a korean satellite kompsat. We present a binary integer programming model for image collection planning. We show the heuristic approach based on the lagrangian relaxation. We present the results on a multi orbit long term planning environment."

FIGURE 5.3: Input is only an abstract from CSPubSum dataset. Highlights produced by the pointer-generator type models with and without NER are shown. Input and author-written research highlights taken <https://www.sciencedirect.com/science/article/pii/S037722171300307X>.

Author-written research highlights: “We present a lightweight non parametric approach to generate wrinkles for 3d facial modeling and animation. Our method represents a convenient approach for generating plausible facial wrinkles with low cost. Our method enables the reconstruction of captured expressions with wrinkles in real time.”
PGM: “We propose a non parametric facial modeling 3d face models from the 3d facial modeling. Synthesize the 3d face expression model with 3d depth camera is considered. Propose a non parametric face method to evaluate the performance of the 3d facial modeling. Method is provided to evaluate the performance of the proposed approach with respect to the existing method.”
PGM + Coverage: “We propose non parametric face acquisition 3d facial modeling models. Face expression model is based on the texture synthesis of multiple subjects. Synthesis method guarantees to 3d face face acquisition. One high quality 3d face model is studied.”
NER + PGM: “A novel synthesis method is proposed to enhance the wrinkles using a single low cost kinect camera. The lightweight feature of the method is that it can generate plausible wrinkles using a single low cost kinect camera and one high quality 3d face model with details as the example. User specific expressions are used as blendshapes to capture facial animations in real time kinect camera and one high quality 3d face model with details.”
NER + PGM + Coverage: “We present a lightweight non parametric method to generate wrinkles for 3d facial modeling and animation. The lightweight feature of the method is that it can generate plausible wrinkles. Our method is low cost and convenient for common users.”

FIGURE 5.4: The input consists of the concatenation of the abstract and conclusion of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without NER are shown. Input and author-written research highlights taken <https://www.sciencedirect.com/science/article/pii/S0010448514001857>.

<p>Author-written research highlights: “We propose a novel parallel 3d delaunay meshing algorithm for large scale simulations. The model information is kept during parallel triangulation process. A 3d local non delaunay mesh repair algorithm is proposed. The meshing results can be very approaching to the model boundary. The method can achieve high parallel performance and perfect scalability.”</p>
<p>PGM: “We propose a solid model boundary preserving method for large scale parallel 3d delaunay meshing. Parallel 3d local mesh 3d delaunay meshing algorithm is proposed. Mesh reconstruction is iteratively performed to meet both the mesh and the shared interfaces. Propose a parallel 3d local mesh reconstruction algorithm to construct delaunay triangulation. Results show high performance and perfect scalability.”</p>
<p>PGM + Coverage: “A new semantic parallel algorithm is proposed for large scale parallel 3d delaunay meshing. Numerical local mesh is the sampling vertices for the problem 3d delaunay meshing. Propose a parallel su based partitioning algorithm by solving the algorithm. Proposed algorithm is highly parallelized to large scale sets and high quality partition walls.”</p>
<p>NER + PGM: “We propose a solid model boundary preserving method for large scale 3d delaunay meshing. The 3d boundary representation model information is kept during the entire parallel 3d delaunay triangulation process. The 3d boundary representation model information is kept during the entire parallel 3d delaunay triangulation process. A parallel 3d local mesh optimization algorithm is presented. Experimental results demonstrate high performance and perfect scalability.”</p>
<p>NER + PGM + Coverage: “We propose a solid model boundary preserving method for large scale parallel delaunay meshing. The 3d boundary representation model information is during the entire parallel 3d delaunay triangulation process. A parallel local mesh refinement algorithm to repair the non delaunay mesh is proposed. A parallel 3d delaunay mesh refinement is presented. Experimental results demonstrate scalability performance.”</p>

FIGURE 5.5: The input consists of the concatenation of the introduction and conclusion of a paper from the CSPubSum dataset. Highlights produced by the pointer-generator type models with and without NER are shown. Input and author-written research highlights taken <https://www.sciencedirect.com/science/article/pii/S0010448514001821>.

5.9 Discussion

In this research, we assessed the performance of four distinct deep neural models for generating research highlights from scientific papers. Among these, the NER-based pointer-generator model combined with a coverage mechanism emerged as the most effective. This model’s strength lies in its capability to integrate named entity recognition (NER) to handle multi-word entities as cohesive and single entity, preventing issues such as fragmentation and misrepresentation of key entities.

The enhanced performance of the NER-based model highlights the importance of accurately identifying and preserving entities within the summarization process. Our findings suggest that incorporating NER can significantly improve the quality of generated research highlights, ensuring that the summaries maintain coherence and relevance.

Future work will focus on investigating the underlying reasons behind the improvement observed with NER integration. This includes exploring how NER contributes to the overall performance and identifying potential areas for further refinement. Additionally, ensuring the syntactic and semantic correctness of the generated highlights remains a crucial objective. Addressing these aspects will enhance the reliability of the summaries and their usefulness for researchers.

5.10 Summary

This chapter highlights the effectiveness of integrating named entity recognition (NER) with deep neural models for generating research highlights from scientific papers. The pointer-generator model enhanced with NER and a coverage mechanism proved to be the most effective in producing highquality highlights. This approach addresses significant challenges in summarization, such as the accurate representation of multi-word entities.

Our results indicate that NER-based techniques can substantially improve the precision and coherence of research summaries. Looking forward, our research will extend to investigating entity hallucination within scientific text summarization. This involves examining how well the summarization models handle and represent named entities, particularly focusing on preventing incorrect or misleading entity generation. This future work aims to further refine summarization methods and ensure the accuracy and relevance of automated research highlights.

6

Hallucination Reduction in Long-Form Text Summarization and Research Highlight Extraction

This chapter investigates the use of deep learning techniques to generate abstracts and research highlights from long scientific papers and measures to ensure the absence of hallucinations. We utilized the Longformer Encoder-Decoder (LED) model for this task, incorporating data filtering and JAENS to assess their effects on factual consistency [161].

6.1 Introduction

Text summarization is a complex task that requires a system to comprehend an entire document and generate a concise, coherent summary that accurately reflects the content of the original text. This process involves two key stages: natural language understanding (NLU) and natural language generation (NLG). With recent advancements in sequence-to-sequence deep learning technologies, particularly transformer-based language models, the capabilities of NLG have significantly improved, leading to more fluent and coherent text generation. However, these advancements have also introduced new challenges, most notably the issue of hallucination, where the system generates

content that is fluent and seemingly natural but factually incorrect or unrelated to the source material.

This chapter addresses the critical issue of hallucination in the context of long-form text summarization and research highlight extraction, focusing on strategies to mitigate this problem and improve the accuracy and reliability of generated summaries in scientific research.

6.2 Background and Motivation

This section provides an overview of the background related to research highlight generation and the motivation behind this research.

6.2.1 Background

The rapid progress in natural language generation has been largely driven by the development of deep learning models such as transformer architecture [195], which have set new benchmarks in tasks like machine translation, text summarization, and conversational AI. These models leverage large scale pre-training on vast datasets, enabling them to generate human-like text. In the context of text summarization, these advancements have allowed for the generation of more coherent and contextually appropriate summaries. However, despite these improvements, a persistent issue remains: the tendency of these models to generate hallucinated content. In NLG, hallucination refers to the generation of text that appears fluent and well-structured but is factually incorrect or inconsistent with the source document. This issue is particularly problematic in abstractive summarization, where the generated summaries are not directly extracted from the source but are instead rephrased and synthesized by the model. As a result, the generated text may include information that was never present in the original document, leading to misleading or erroneous summaries. Large pre-trained transformer models have proven to be exceptionally capable of dealing with natural language tasks [50, 160]. Handling extended textual sequences, on the other hand, remains a considerable issue for these pre-trained transformer models. These challenging input documents are often substantially longer than the maximal context lengths of typical transformer models, necessitating both specialized model architectural adjustments and unique training regimes to accommodate. For example, numerous memory-efficient transformer variations have been proposed to prevent the quadratic escalation in memory consumption of the attention estimation in transformers. Another severe issue is the inability of current abstractive summarization methods to generate faithful results. These systems frequently struggle to verify that the generated summaries only include information

extracted from the source document and do not include misinterpreted or hallucinated statements. These hallucinations can occur for a variety of causes, including biases in the training data, a lack of context perception, or model over-optimization. Cao et al. [34] and Kryściński et al. [90] reported that approximately 30% of the summaries generated by seq2seq models suffer from the issue of hallucination. As a result, as noted in the works of the NLP community, attention has been drawn more and more to the faithfulness and factual components of abstractive summarization [90, 63, 230]. Many recent works study entity-level and relation-level hallucination problems in the generated text. Nan et al. [130] address entity hallucination by applying a filter on the training data and multi-task learning. Goyal and Durrett [63] study relation hallucination, that is, whether the semantic relationships manifested by the individual dependency arcs in a generated sentence are entailed by the source sentence. One notable work by Narayan et al. [131] incorporates entity chain content planning to guide faithful summary generation. There has been growing interest in quantitatively measuring the faithfulness of text generation models. Most widely-adopted evaluation metrics for text generation, such as ROUGE [102] and BERTScore [224] correlate poorly with the human perceived faithfulness of the generated text [90]. Recent studies explore categorical and content-based analysis for measuring the faithfulness of summaries [63].

6.2.2 Motivation

The motivation for addressing hallucination in abstractive text summarization stems from the increasing reliance on automated systems to generate summaries in various domains, particularly in scientific research. In fields where accuracy and factual consistency are paramount – such as medicine, law, and academia, the presence of hallucinated content in summaries can have severe consequences. For instance, in the context of scientific research, a summary that inaccurately reflects the content of a research paper could mislead readers and undermine the credibility of the research. Similarly, in medical domain, an inaccurate summary generated from patient records could lead to incorrect diagnoses or treatment plans, posing significant risks to patient safety. The need to ensure that automated summarization systems produce reliable and factually consistent summaries is, therefore, of utmost importance. This chapter explores methods to reduce hallucination in long-form text summarization, with the aim of enhancing the trustworthiness and utility of these systems in important applications.

6.3 Challenges and Opportunities

This section addresses the primary difficulties associated with creating precise research highlights and summaries, with a particular focus on the problem of hallucination in long-form text summarization. Additionally, it explored potential advancements through methods and technologies designed to enhance the accuracy and dependability of automated summarization systems.

6.3.1 Challenges

One of the most significant challenges in developing reliable text summarization systems is the problem of hallucination, particularly in the domain of abstractive summarization. Despite the progress made with pre-trained language models (PLMs) and large language models (LLMs), ensuring that the generated summaries are factually consistent with the source document remains a difficult task. Hallucinations in summarization can occur for several reasons, including the model's reliance on patterns learned from training data that may not always align with the specific content of a given document. Additionally, the complexity of the source material, especially in scientific research papers, can exacerbate the problem, as the model may struggle to accurately condense and rephrase technical information without introducing errors. The challenge is further intensified by the need to balance brevity and informativeness in the generated summaries, which can sometimes lead to the omission of critical details or the inclusion of incorrect information.

6.3.2 Opportunities

Addressing the hallucination problem in text summarization presents a significant opportunity to improve the effectiveness and reliability of the natural language generation systems, particularly in high-stakes domains such as scientific research, healthcare, and law. By developing and implementing advanced **filtering techniques** and other mitigation strategies, it is possible to reduce the incidence of hallucination and enhance the factual consistency of generated summaries. Addressing hallucination not only improves the trustworthiness of automated summarization systems but also expands their applicability in domains where accuracy is critical. Furthermore, the insights gained from reducing hallucination in abstractive text summarization can be applied to other NLG tasks, contributing to the broader goal of creating more reliable and user-aligned AI systems. In this chapter, we will explore various approaches to reducing hallucination in the context of scientific research paper summarization and research highlight extraction, with the aim of producing more accurate and reliable summaries that meet

the needs of users in these demanding fields.

6.4 Main Contributions

The main contributions of this chapter are:

1. We used Longformer Encoder-Decoder (LED) model [22] to generate summary of scientific articles in the PubMed dataset [41]. We also extracted research highlights from the CSPubSum dataset [43]. In addition, we explored two techniques, namely, data filtering and JAENS (Join sAlient ENtity and Summary generation) [130] to study their effect on the factual consistency of the generated summaries for both datasets.
2. We analyzed the factual consistency of the output summary at the entity level using the following metrics: precision-source, precision-target, recall-target, and F1-target, introduced by [130]. We also used the traditional metrics, namely, ROUGH [102], METEOR [17], MoverScore [227], and BERTScore [224], to evaluate the performance of the models.

6.5 Methodology

To handle long input sequences, we utilized the pre-trained checkpoints of the Longformer Encoder Decoder (LED) model [22], which incorporates a sliding window and dilated sliding window attention mechanisms. It consists of both the encoder and decoder Transformer stacks, but instead of using full self-attention in the encoder, it employs the Longformer’s efficient local+global attention pattern. The decoder applies full self-attention to all encoded tokens and previously decoded locations. Because pre-training LED is expensive, authors in [22] have used BART parameters to initialize LED parameters and adhered to BART’s exact design in terms of the number of hidden sizes and layers. This allows it to effectively process lengthy inputs. We performed fine-tuning of the pre-trained LED model to adapt it specifically for text summarization of scientific documents. To ensure the accuracy of the summaries, we implemented scispaCy-based Named Entity Recognition (NER) on the ground truth summaries. We applied the JAENS (Join sAlient ENtity and Summary generation) approach to augment salient entities in front of the abstracts. Training the model to recognize summary-worthy named-entities aims to enhance the precision and recall related to named-entities in the generated summaries.

We have performed experiments with 3 variants with the LED model: (1) fine-tuned on the LED model, (2) fine-tuned LED model with the filtered dataset, and (3)

fine-tuned LED model using the JAENS approach on the filtered dataset.

6.5.1 Fine-tuning LED

Pre-trained models like LED learn rich language representations from a large corpus. Fine-tuning customizes these models for specific tasks. It initializes the model with pre-trained weights, then fine-tunes it on a task-specific dataset using backpropagation. Fine-tuning leverages the model’s language understanding saves time and resources, and requires less labeled data. This approach enhances text summarization by adapting the model to task-specific data while leveraging its pre-trained knowledge.

6.5.2 Entity-based Data Filtering

As demonstrated successfully by [130], the training dataset’s quality has a significant impact on the amount of entity-level hallucinations present in the generated summary. With that in mind, we applied scispaCy Named Entity Recognition (NER) to the gold summary for the PubMed dataset. This allows us to identify all the named-entities present in the gold summary. Our objective is to ensure that these named-entities have corresponding n -gram matches within the source document. For unigram matching, we avoid matching any stop words. Therefore, if any named-entity of a sentence in the summary cannot be found within the source document, we decided to exclude that sentence from the summary. If the number of sentences in the summary is one and using the filtering technique we need to remove that sentence, then the entire article-summary pair has been removed from the dataset.

6.5.3 Join sAlient ENtity and Summary Generation (JAENS)

The JAENS (Join sAlient ENtity and Summary generation) approach, originally introduced by Nan et al. [130], is an alternative generative approach aimed at enhancing entity-level precision, and recall metrics. JAENS trains the LED model to construct a sequence that contains summary-worthy named-entities, a special token, and the summary itself, as opposed to typical summarization approaches. This approach enables the model to simultaneously learn the identification of summary-worthy named-entities while generating summaries, similar to the multitask learning approach. By prioritizing the generation of salient named-entities in the decoder, JAENS ensures that the summaries incorporate and highlight these important entities through decoder self-attention. By incorporating the JAENS approach into our project, we aim to mitigate entity-level summary hallucinations and improve the overall quality of the generated summaries.

6.6 Experimental Setup

In this section, we discussed the datasets used, the data pre-processing steps, the implementation details, and some automatic evaluation metrics that are not explored in Section 2.7 of the second chapter.

6.6.1 Dataset Details

We used two datasets: PubMed [41] and CSPubSum [43].

We used a dataset collected from a scientific repository, PubMed¹, and which was introduced in [41]. We chose scientific papers as our dataset because they are examples of long documents with a standard discourse structure. Furthermore, scientific papers are rich in domain-specific terminology and technical information, which makes them an important source of information for researchers and practitioners alike. PubMed is a biomedical literature database that contains over 30 million citations and abstracts of research articles. The dataset contains almost 19,000 scholarly publications on diabetes from the PubMed database, which are categorized into one of three categories.

In our experiment, we chose, for training 2000 examples, validation 250 examples, and testing 250 examples. The size of the used dataset after applying the entity-based filtering procedure was 1798 examples for training, 232 examples for validation, and 236 examples for testing. The average number of sentences in summary before applying the entity-based data filtering technique was 7.33, 7.04, and 7.51 for training, validation, and test datasets. The average number of sentences in a summary after applying the entity-based data filtering technique is 4.34, 4.11, and 4.58 for training, validation, and test datasets.

For CSpubSum, we started with 8120 training, 1014 validation, and 1013 testing examples. After filtering, the sizes were 7297 for training, 932 for validation, and 913 for testing. The average number of sentences in the author-written research highlights before applying the entity-based data filtering technique was 4.23, 4.20, and 4.19 for training, validation, and test datasets. The average number of sentences in the author-written research highlights after applying the entity-based data filtering technique is 4.26, 4.22, and 4.21 for training, validation, and test datasets.

6.6.2 Data Processing

PubMed: For the PubMed dataset, we eliminated all punctuation, numerals, special characters, mathematical formulas, and citation markers from the documents and lowercase the entire corpus. When we were going through documents, we made sure

¹<https://pubmed.ncbi.nlm.nih.gov/>

they were the right length and had the right structure. If something was too long, like a thesis, or too short, like a tutorial announcement, we removed it. We also looked for documents that did not have an abstract or a clear structure. To understand the structure, we used the section headings as clues. Sometimes, documents had figures or tables that did not help us understand the text. We got rid of those, keeping only the words. We reorganized the dataset into pairs of article and abstract. In our model, the maximum number of allowed input tokens is 8192, that of output tokens is 512, and the minimum number of output tokens is 100 only. In line with the JAENS approach, we used the scispaCy model `en_core_sci_sm`² library to generate summary-worthy named-entities and augmented the list of comma-separated named-entities before the ground truth summary (abstract) for each sample of the dataset. The sequence of named-entities is followed by a special token, which helps separate the entities from the abstract. This special token is chosen from the model’s vocabulary such that it is not commonly occurring and can help the model learn to recognize the named-entities separately from the actual abstract. This helps in training the model as now the model will apply special attention to these entities while generating the summary.

CSPubSum: For the CSPubSum dataset, we only removed citation markers and blank spaces. We reorganized the dataset into pairs of abstracts and author-written research highlights. For the CSPubSum dataset, we predicted research highlights from abstracts, with a maximum of 1500 input tokens and 512 output tokens, and a minimum of 10 output tokens. In line with the JAENS approach, we utilized the same scispaCy model `en_core_sci_sm` to generate summary-worthy named entities for the CSPubSum dataset. We then augmented each sample by adding a list of comma-separated named entities before the ground-truth summary (research highlights). To separate the named entities from the research highlights, we included a special token from the model’s vocabulary that is rarely used. This special token helps the model in distinguishing the named entities from the actual research highlights, allowing the model to focus more on these entities during summary generation.

6.6.3 Implementation Details

We conducted our experiments using Google Colab Pro+, which provided us with an NVIDIA A100 GPU. For all experiments, we used the base variant of the pre-trained LED model `led-base-16384`³, due to resource limitations.

PubMed Dataset : Firstly, we fine-tuned the LED model on the original 2000-sample PubMed dataset. Secondly, we utilized a filtered version of the taken dataset by removing article-abstract pairs with a $prec_s$ score (to be defined in the next sub-section

²<https://allenai.github.io/scispacy/>

³<https://huggingface.co/allenai/led-base-16384>

6.6.4) less than 1 (i.e., we ensure that the abstract – which is the ground-truth summary – contains almost no hallucinations of entities) and performed fine-tuning on this filtered dataset. Finally, we incorporated the JAENS approach into the fine-tuning process by augmenting summary-worthy named-entities in front of the abstract for each example of the filtered train dataset, aiming to enhance entity-level precision, recall, and F1 metrics in the generated summaries and thus reduce the entity-level hallucinations. For all the models, we fine-tuned up to 10 epochs. To evaluate all three models, we used the same test dataset that was obtained after the entity-based data filtering technique.

CSPubSum Dataset : Firstly, we fine-tuned the LED model on the original 8120-sample CSPubSum dataset. Secondly, we utilized a filtered version of this dataset by removing abstract-author written research highlights pairs with a $prec_s$ score (to be defined in the next sub-section 6.6.4) less than 1, ensuring that the author written research highlights — serving as the ground-truth summaries — contained minimal hallucinations of entities. We then performed fine-tuning on this filtered dataset. Finally, we integrated the JAENS approach into the fine-tuning process by augmenting summary-worthy named entities in front of the research highlights for each example in the filtered training dataset. This approach aimed to improve entity-level precision, recall, and F1 metrics in the generated summaries, thus reducing entity-level hallucinations. For all models, we fine-tuned up to 5 epochs. To evaluate all three models, we used the same test dataset obtained after the entity-based data filtering technique.

For both the PubMed and CSPubSum datasets, we used the same hyperparameters for all models. Specifically, `num_beams` was set to 4, `length_penalty` was 2.0, `early_stopping` was enabled, and `no_repeat_ngram_size` was set to 3. The metric used for selecting the best model was the ROUGE-1 F-measure.

6.6.4 Evaluation Metrics

We used a comprehensive set of widely used evaluation metrics for text summarization models, including ROUGE [102], METEOR [17], MoverScore [227], and BERTScore [224], to assess the quality and effectiveness of the generated summaries. Unfortunately, these metrics are inadequate to quantify factual consistency [90]. Hence, we have also used three new metrics, introduced by [130], to evaluate the factual consistency of the generated summaries.

We define $\mathcal{N}(t)$ as the count of named-entities in the target (ground truth or gold summary) and $\mathcal{N}(h)$ as the count of named-entities in the hypothesis (generated summary). To determine the number of entities in the hypothesis that have corresponding matches in the source document, we use $\mathcal{N}(h \cap s)$. In circumstances when a named-entity in the summary spans many words, we consider it a match if any component of the

named-entity can be identified in the original document, permitting partial matching based on n -grams. **Precision-source**, as defined in Equation 6.1,

$$prec_s = \mathcal{N}(h \cap s) / \mathcal{N}(h) \quad (6.1)$$

It is a metric that is used to determine the intensity of hallucination in relation to the source. Note that $prec_s$ represents the percentage of entities mentioned in the generated summary that can be retrieved from the source. Low $prec_s$ indicates that hallucination is possibly present in the generated text. However, $prec_s$ does not capture the computed summary’s entity-level correctness in relation to the ground-truth summary. The entity-level accuracy of the generated summary is calculated using the **precision-target** and **recall-target**, as defined in Equations 6.2 and 6.3:

$$prec_t = \mathcal{N}(h \cap t) / \mathcal{N}(h) \quad (6.2)$$

$$recall_t = \mathcal{N}(h \cap t) / \mathcal{N}(t) \quad (6.3)$$

The expression for calculating the **F1 score** is given in the Equation 6.4.

$$F1_t = \frac{2 * (recall_t * prec_t)}{recall_t + prec_t}. \quad (6.4)$$

Here, $\mathcal{N}(h \cap t)$ represents the number of matched named-entities in the generated summary and the ground truth summary.

Note that the above precision and recall scores can be calculated in two ways. One is to consider the entity mentioned in each document (which may be the source s or target t or hypothesis h) as a set so that multiple occurrences of an entity in a document are equivalent to a single occurrence. The other is to consider the entity mentioned in a document as a list; here, if a metric is defined as $\mu = \mathcal{N}(x \cap y) / \mathcal{N}(x)$, then for each entity mention in x , we check if it occurs in y , and if so, increment the intersection count $\mathcal{N}(x \cap y)$ by unity. The second approach is followed in [130]. In the first approach, we denote the metrics as $prec_s^U$, $prec_t^U$, $recall_t^U$, and $F1_t^U$ (U indicates that only unique entity mentions are considered). In the second, we represent them as $prec_s^{NU}$, $prec_t^{NU}$, $recall_t^{NU}$, and $F1_t^{NU}$.

6.7 Results

6.7.1 Results on PubMed Dataset

Table 6.1 shows the F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, METEOR, Mover-Score, and BERTScore metrics on the filtered test dataset of PubMed. Evaluation

with entity-level factual consistency metrics precision-source ($prec_s^{NU}, prec_s^U$), precision-target ($prec_t^{NU}, prec_t^U$), recall-target ($recall_t^{NU}, recall_t^U$), and F1-target ($F1_t^{NU}, F1_t^U$) is shown in Table 6.2.

In Table 6.1, the LED model fine-tuned without additional techniques like filtering or JAENS shows the highest values for ROUGE, METEOR, MoverScore, and BERTScore. This shows that not only n -gram matches and cosine similarity of embeddings are higher for the plain LED model. However, when fine-tuned with filtering technique, the LED model achieved the highest precision-source ($prec_s^{NU}, prec_s^U$) as shown in Table 6.2. Applying filtering and the JAENS technique during LED fine-tuning enhances precision-target metrics. This indicates that these methods help the model make more accurate predictions for entities, reducing the number of false positives and improving the precision-target ($prec_t^{NU}, prec_t^U$) metric results shown in Table 6.2. In terms of recall-target ($recall_t^{NU}, recall_t^U$), the LED model fine-tuned without any filtering or JAENS technique achieved the highest score. This means that while additional techniques improve precision, they may also lead to a decrease in the model’s ability to identify all relevant positive instances. Thus, fine-tuning the LED model without additional techniques like filtering and JAENS resulted in the highest F1-target ($F1_t^{NU}, F1_t^U$) metric values. Nan et al. [130] also observed a reduction in ROUGE scores when data filtering and JAENS were applied, and remarked that it could be due to the increased complexity during decoding. Surprisingly, we find that data filtering and JAENS do not improve the $F1_t$ scores. In future, we intend to conduct a detailed study of this behaviour and try to decipher its reason. This could be related to the inaccuracy in entity recognition that we observed for the dataset; for example, on manual review, we found that many phrases detected as entities do not appear to be very important, but their match/mismatch between the generated and golden summary do impact the $F1_t$ scores. In contrast, in [130], standard entities are detected which could be achieved with high accuracy. Another difference with [130] is that in our case, the dataset is much smaller and the summaries longer.

TABLE 6.1: Evaluation of variants of LED fine-tuned models with and without filtering and JAENS technique: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on inputs that are the article body from the PubMed dataset. All scores are presented as percentages (%).

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSum	METEOR	MoverScore	BERTScore
Fine-tuned LED	35.12	14	21.57	29.96	32.08	18.25	84.96
Fine-tuned LED + Filtered dataset	33.18	12.04	19.93	28.48	27.43	16.23	84.74
Fine-tuned LED + Filtered dataset + JAENS	30.21	09.13	18.26	25.87	23.55	14.07	84.35

TABLE 6.2: Evaluation of the variants of LED fine-tuned models with and without filtering and JAENS technique: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated summaries for the PubMed dataset. All scores in percentage (%).

Model Name	$prec_s^{NU}$	$prec_s^U$	$prec_t^{NU}$	$recall_t^{NU}$	$F1_t^{NU}$	$prec_t^U$	$recall_t^U$	$F1_t^U$
Fine-tuned LED	94.76	93.38	39.91	60.98	46.14	39.91	54.74	43.76
Fine-tuned LED + Filtered dataset	96.83	96.04	40.86	52.03	43.27	40.86	45.36	40.15
Fine-tuned LED + Filtered dataset + JAENS	92.16	89.36	43.28	43	40.15	43.28	36.36	36.34

6.7.2 Results on CSPubSum Dataset

Table 6.3 shows the F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore, and BERTScore metrics on the filtered test dataset of CSPubSum. Evaluation with entity-level factual consistency metrics precision-source ($prec_s^{NU}$, $prec_s^U$), precision-target ($prec_t^{NU}$, $prec_t^U$), recall-target ($recall_t^{NU}$, $recall_t^U$), and F1-target ($F1_t^{NU}$, $F1_t^U$) is shown in Table 6.4. In Table 6.3, the LED model fine-tuned with additional techniques like filtering shows the highest values for ROUGE, MoverScore, and BERTScore. The fine-tuned LED model only outperforms on METEOR score values. But, when fine-tuned the LED model without any additional techniques like filtering and JAENS achieved the highest precision-source ($prec_s^{NU}$, $prec_s^U$) as shown in Table 6.4. Applying filtering and the JAENS technique during LED fine-tuning enhances precision-target metrics. This indicates that these methods help the model make more accurate predictions for entities, reducing the number of false positives and improving the precision-target ($prec_t^{NU}$, $prec_t^U$) metric results shown in Table 6.4. In terms of recall-target ($recall_t^{NU}$, $recall_t^U$), the LED model fine-tuned without any filtering and JAENS technique achieved the highest score. This means that while additional techniques improve precision, they may also lead to a decrease in the model’s ability to identify all relevant positive instances. Thus, fine-tuning the LED model without additional techniques like filtering and JAENS resulted in the highest F1-target in terms of $F1_t^{NU}$ metric values. But, fine-tuning the LED model with additional techniques like filtering resulted in the highest F1-target in terms of $F1_t^U$ metric values.

TABLE 6.3: Evaluation of variants of LED fine-tuned models with and without filtering and JAENS technique: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on inputs that are the abstract only from the CSPubSum dataset. All scores are presented as percentages (%).

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-LSum	METEOR	MoverScore	BERTScore
Fine-tuned LED	37.46	15.21	25.82	25.82	36.74	23.21	87.7
Fine-tuned LED + Filtered dataset	38.05	15.44	26.43	26.43	36.48	23.93	87.87
Fine-tuned LED + Filtered dataset + JAENS	35.33	11.99	24.65	24.65	29.4	22.52	87.68

TABLE 6.4: Evaluation of the variants of LED fine-tuned models with and without filtering and JAENS technique: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated summaries for the CSPubSum dataset. All scores in percentage (%).

Model Name	$prec_s^{NU}$	$prec_s^U$	$prec_t^{NU}$	$recall_t^{NU}$	$F1_t^{NU}$	$prec_t^U$	$recall_t^U$	$F1_t^U$
Fine-tuned LED	99.31	99.04	46.88	67.98	53.09	41.63	59.33	46.96
Fine-tuned LED + Filtered dataset	99.16	98.79	47.99	65.23	53.07	43.0	57.67	47.41
Fine-tuned LED + Filtered dataset + JAENS	94.72	93.07	50.82	48.25	47.88	45.81	42.72	42.78

6.8 Case Studies

In this section, we present a few examples demonstrating the outputs produced by variants of the fine-tuned LED model, with and without additional techniques like filtering and JAENS. In all case studies reported below, **yellow** represents an incorrect representation of an entity during summary generation by the fine-tuned LED model. **cyan** denotes a correctly generated entity by the model using additional techniques with the fine-tuned LED that was incorrectly represented by the fine-tuned LED model without any additional techniques. Figure 6.1 shows sample outputs generated by fine-tuning the LED model with the PubMed dataset without filtering, with the filtered dataset, and with both the filtered dataset and JAENS technique. The entities detected in each summary are also shown. The original abstract consists of 5 sentences, but after using the entity-based filtering technique, it consists of only 2 sentences. Similarly, Figures 6.2 and 6.3 show sample outputs generated by fine-tuning the LED model with the CSPubSum dataset without filtering, with the filtered dataset, and with both the filtered dataset and JAENS techniques. The entities detected in each summary are also shown.

Figure 6.1 shows that the fine-tuned LED model incorrectly generated entities or

phrases like “botic stress responses”, “methods”, and “plant cwi”. In contrast, the fine-tuned LED model with additional techniques like filtering and JAENS correctly generated entities or phrases such as “plant cwi maintenance mechanism” and “cell wall integrity (cwi) maintenance”. Figure 6.2 shows that the fine-tuned LED model incorrectly generated entities or phrases like “BVDODE”, and “BVBODE”. In contrast, the fine-tuned LED model with additional techniques like filtering and JAENS correctly generated entities or phrases such as “BVODE”. Figure 6.3 shows that the fine-tuned LED model incorrectly generated entities like flexible linear regression (LR) models” instead of “flexible logistic regression (LR) models”. However, the fine-tuned LED model with additional techniques like filtering and JAENS correctly generated entities such as “flexible LR model”, though it still did not generate the full form “logistic regression”. This is still an improvement over the incorrect generation of “logistic regression”.

<p>Ground-truth summary (abstract): “in recent years, evidence has accumulated that a dedicated plant cell wall integrity (cwi) maintenance mechanism exists. the available data suggest that it may represent a component of the stress response mechanisms underlying biotic and abiotic stress responses, which has not been identified previously as a distinct mechanism.”</p> <p><i>entities=‘years’, ‘evidence’, ‘plant’, ‘cell wall’, ‘integrity’, ‘cwi’, ‘maintenance’, ‘mechanism’, ‘data’, ‘stress response’, ‘mechanisms’, ‘biotic’, ‘abiotic stress responses’</i></p>
<p>Fine-tuned LED: “plant cells need to maintain the functional integrity of their walls during cell morphogenesis and exposure to biotic / abiotic stress. the available evidence suggests that a dedicated plant cell wall integrity (cwi) maintenance mechanism exists. while our understanding of the mechanisms regulating stress responses and morphogenesis has increased significantly, our knowledge regarding the processes maintaining cwi is still limited. in the last years, a number of yeast strains have been identified and the available knowledge regarding cwi maintenance during plant development and in yeast has increased substantially. in this review, we will focus on the role of particular cell wall defects in the perception of and response to botic stress responses .</p> <p>materials and methodsa yeast cwi monitoring and maintenance network is quite complex, providing an indication of the possible complexity of the interaction and the direct impact of these differences on the biological role and function of the plant cwi .”</p> <p><i>entities=‘plant cells’, ‘functional’, ‘integrity’, ‘walls’, ‘cell morphogenesis’, ‘exposure to’, ‘biotic’, ‘abiotic stress’, ‘evidence’, ‘plant’, ‘cell wall’, ‘cwi’, ‘maintenance’, ‘mechanism’, ‘mechanisms’, ‘stress responses’, ‘morphogenesis’, ‘increased’, ‘knowledge’, ‘processes’, ‘years’, ‘yeast’, ‘strains’, ‘identified’, ‘plant development’, ‘review’, ‘defects’, ‘perception’, ‘response’, ‘botic stress’, ‘methods’, ‘monitoring’, ‘network’, ‘quite complex’, ‘indication’, ‘complexity’, ‘interaction’, ‘impact’, ‘biological role’, ‘function’, ‘plant cwi’</i></p>
<p>Fine-tuned LED + Filtered dataset: “ plant cells need to maintain the functional integrity of their walls during cell morphogenesis and exposure to biotic / abiotic stress responses. the available evidence suggests that a dedicated plant cell wall integrity (cwi) maintenance mechanism exists. while the yeast cwi monitoring and maintenance network is quite complex, certain important differences exist that affect the biological role and function of the plant cwi maintenance mechanism . in this review, we will focus on the effects of particular cell wall defects on pathogen resistance and the mode of action of specific signaling cascades.”</p> <p><i>entities=‘plant cells’, ‘functional’, ‘integrity’, ‘walls’, ‘cell morphogenesis’, ‘exposure to’, ‘biotic’, ‘abiotic stress responses’, ‘evidence’, ‘plant’, ‘cell wall’, ‘cwi’, ‘maintenance’, ‘mechanism’, ‘yeast’, ‘monitoring’, ‘network’, ‘quite complex’, ‘biological role’, ‘function’, ‘review’, ‘effects’, ‘defects’, ‘pathogen resistance’, ‘mode’, ‘signaling cascades’</i></p>
<p>Fine-tuned LED + Filtered dataset + JAENS: “plant cells need to maintain the functional integrity of their walls during cell morphogenesis and exposure to biotic / abio stress responses. the available evidence suggests that a dedicated plant cwi maintenance mechanism exists, but what about the role of dedicated cells wall integrity (cwi) maintenance during process.”</p> <p><i>entities=‘plant cells’, ‘functional’, ‘integrity’, ‘walls’, ‘cell morphogenesis’, ‘exposure to’, ‘biotic’, ‘abio stress responses’, ‘evidence’, ‘plant’, ‘maintenance’, ‘mechanism’, ‘cells’, ‘cwi’, ‘process’</i></p>

FIGURE 6.1: Comparison of ground-truth and generated summary from the test set of the PubMed dataset. Hallucinations are highlighted. Input and ground-truth summary (abstract) taken from <https://pubmed.ncbi.nlm.nih.gov/22629279/>.

<p>Author-written research highlights: “We solve numerically a nonlinear boundary-value problem modeling corneal topography. We use the method of lines to facilitate fast computation and include the R routines. We derive some estimates which demonstrate the order of convergence of the algorithm.”</p> <p><i>entities= ‘corneal topography’, ‘method’, ‘fast computation’, ‘R routines’, ‘estimates’, ‘convergence’, ‘algorithm’</i></p>
<p>Fine-tuned LED: “A nonlinear, two-point boundary value ordinary differential equation (BVODE) that defines corneal curvature according to a static force balance is introduced. A numerical solution to the BVODE is computed by first converting the BVDODE to a parabolic partial differential equation. A pseudo-time derivative is added to the PDE by adding an initial value (t, pseudo-Time) derivative to the bVODE. The PDE solution at this point is also the solution for the BVBODE.”</p> <p><i>entities= ‘ordinary differential equation’, ‘BVODE’, ‘corneal curvature’, ‘static force’, ‘balance’, ‘numerical solution’, ‘BVDODE’, ‘parabolic partial’, ‘differential equation’, ‘pseudo-time derivative’, ‘PDE’, ‘initial value’, ‘bVODE’, ‘solution’, ‘BVBODE’</i></p>
<p>Fine-tuned LED + Filtered dataset: “A nonlinear, two-point boundary value ordinary differential equation (BVODE) that defines corneal curvature according to a static force balance is introduced. A numerical solution to the BVODE is computed by the method of lines (MOL) with the calculation of t such that the derivative in t reduces to essentially zero. An estimate for the cornea radius of curvature is also derived.”</p> <p><i>entities= ‘ordinary differential equation’, ‘BVODE’, ‘corneal curvature’, ‘static force’, ‘balance’, ‘numerical solution’, ‘method’, ‘MOL’, ‘calculation’, ‘derivative’, ‘reduces’, ‘zero’, ‘estimate’, ‘cornea radius’, ‘curvature’</i></p>
<p>Fine-tuned LED + Filtered dataset + JAENS: “A numerical solution is computed to the BVODE. The PDE is also the solution for the PDE. A numerical solution to the ODE is computed by the MOL method. A series approximation is derived from which an estimate for the Cornea radius of curvature is obtained.”</p> <p><i>entities= ‘numerical solution’, ‘BVODE’, ‘PDE’, ‘solution’, ‘ODE’, ‘MOL method’, ‘series approximation’, ‘estimate’, ‘Cornea radius’, ‘curvature’</i></p>

FIGURE 6.2: Comparison of author-written and generated research highlights from the test set of the CSPubSum dataset. Hallucinations are highlighted. Input and author-written research highlights taken from <https://www.sciencedirect.com/science/article/pii/S0010482514001681>.

<p>Author-written research highlights: “We aimed to predict pediatric surgical morbidity using preoperative characteristics. We compared logistic regression models to data mining algorithms. The data mining algorithms performed as well as a simple logistic regression model. A flexible logistic regression model performed best on most model fit criteria.”</p> <p><i>entities=‘pediatric’, ‘surgical morbidity’, ‘preoperative’, ‘characteristics’, ‘logistic regression models’, ‘data mining’, ‘algorithms’, ‘logistic regression model’, ‘flexible’, ‘model’, ‘criteria’</i></p>
<p>Fine-tuned LED: “We compare the performance of flexible linear regression (LR) models for predicting 30-day surgical morbidity in children. We used the 2012 National Surgical Quality Improvement Program-Pediatric dataset to compare LR models. The flexible LR model outperformed the simple LR model in terms of accuracy, sensitivity, specificity, PPV, and NPV. Support vector machines, random forests, and boosted classification trees do not show better performance than flexible LR for predicting pediatric morbidity. After further validation, the flexible model could assist with clinical decision-making based on patient-specific surgical risks.”</p> <p><i>entities=‘compare’, ‘performance’, ‘flexible linear regression’, ‘models’, ‘predicting’, ‘surgical morbidity’, ‘children’, ‘National Surgical Quality Improvement’, ‘Program-Pediatric dataset’, ‘LR models’, ‘flexible’, ‘LR model’, ‘accuracy’, ‘sensitivity’, ‘specificity’, ‘PPV’, ‘NPV’, ‘Support vector machines’, ‘random forests’, ‘classification’, ‘trees’, ‘LR’, ‘pediatric’, ‘morbidity’, ‘validation’, ‘flexible model’, ‘assist’, ‘clinical decision-making’, ‘patient-specific’, ‘surgical’, ‘risks’</i></p>
<p>Fine-tuned LED + Filtered dataset: “We used the 2012 National Surgical Quality Improvement Program-Pediatric dataset to compare the performance of a simple LR model incorporating restricted cubic splines and interactions (simple LR model) and a flexible LR model (flexible LR model). We used ensemble-based methods to compare LR to several data mining algorithms for predicting 30-day surgical morbidity in children. Results showed significantly higher accuracy, sensitivity, specificity, PPV, and NPV than the simple LR models. The flexible LR models performed better in terms of the aforementioned measures or in model calibration or discrimination. The support vector machines, random forests, and boosted classification trees do not show better performance than the flexible LR for predicting pediatric morbidity. After further validation, the flexible model could assist with clinical decision-making based on patient-specific surgical risks.”</p> <p><i>entities=‘National Surgical Quality Improvement’, ‘Program-Pediatric dataset’, ‘compare’, ‘performance’, ‘LR model’, ‘cubic splines’, ‘interactions’, ‘flexible’, ‘ensemble-based methods’, ‘LR’, ‘data mining’, ‘algorithms’, ‘predicting’, ‘surgical morbidity’, ‘children’, ‘Results’, ‘accuracy’, ‘sensitivity’, ‘specificity’, ‘PPV’, ‘NPV’, ‘LR models’, ‘measures’, ‘model calibration’, ‘discrimination’, ‘support vector machines’, ‘random forests’, ‘classification’, ‘trees’, ‘pediatric’, ‘morbidity’, ‘validation’, ‘flexible model’, ‘assist’, ‘clinical decision-making’, ‘patient-specific’, ‘surgical’, ‘risks’</i></p>
<p>Fine-tuned LED + Filtered dataset + JAENS: “We compared the performance of two methods for predicting surgical morbidity in children. We used the 2012 National Surgical Quality Improvement Program-Pediatric dataset to compare the flexible and flexible LR models. We found that none of the models performed better than the flexible model in terms of the aforementioned measures or in model calibration or discrimination.”</p> <p><i>entities=‘performance’, ‘methods’, ‘predicting’, ‘surgical morbidity’, ‘children’, ‘National Surgical Quality Improvement’, ‘Program-Pediatric dataset’, ‘compare’, ‘flexible’, ‘LR models’, ‘models’, ‘flexible model’, ‘measures’, ‘model calibration’, ‘discrimination’</i></p>

FIGURE 6.3: Comparison of author-written and generated research highlights from the test set of the CSPubSum dataset. Hallucinations are highlighted. Input and author-written research highlights taken from <https://www.sciencedirect.com/science/article/pii/S0010482514003266>.

6.9 Discussions

In this study, we worked on reducing hallucinations in summaries of long scientific documents. Hallucinations are when a summary includes information that is not actually in the original text, which can make the summary less reliable. We used the Longformer Encoder-Decoder (LED) model, which we fine-tuned on two datasets: PubMed and CSPubSum. PubMed includes biomedical papers, while CSPubSum focuses on computer science research papers.

We tried two main techniques to improve the quality of the summaries: *data filtering* and *Join sAlient ENtity and Summary generation (JAENS)*. Data filtering involved cleaning up the input data to remove unnecessary or irrelevant information. The *JAENS* technique aimed to generate summaries while also identifying key entities, like important terms or concepts, at the same time.

Our results showed that the LED model performed well on standard metrics like ROUGE, METEOR, MoverScore, and BERTScore on PubMed datasets, while in CSPubSum dataset fine-tuning with LED with filtering techniques performed well except METEOR metrics. However, when we added data filtering, we noticed improvements in some metrics related to factual accuracy, especially *precision-source* and *precision-target* and sometimes in terms of **F1-target**. These improvements were more significant with the PubMed dataset, likely because the biomedical papers are more structured and consistent. The CSPubSum dataset showed smaller improvements in terms of *precision-target* only and sometimes on *F1-target*, which suggests that different types of documents may need different approaches to reduce hallucinations.

Interestingly, the JAENS technique did not always improve the results and sometimes even performed slightly worse than the standard approach. This suggests that combining entity recognition with summary generation might be more complicated than expected, and it needs further investigation. The specific challenges could be due to the way different types of documents are written and how entities are used in them.

We also found that the model struggled to consistently identify and keep accurate entities in the summaries. This suggests there might be weaknesses in the *entity recognition* part of the model, especially when dealing with complex or specialized terms found in scientific papers. Improving this part of the model could help make the summaries more accurate and reduce hallucinations.

This study highlights the importance of using customized techniques to improve summarization performance, especially for lengthy and intricate scientific documents. Through a comparison of results from the PubMed and CSPubSum datasets, key insights were obtained regarding the effectiveness of these methods across different research papers, emphasizing their versatility and positive influence on summarization

accuracy.

6.10 Conclusion

We applied the Longformer Encoder-Decoder model on scientific research papers to generate summaries, leveraging data filtering and the JAENS approach to reduce entity hallucinations. Our results demonstrate that a simple fine-tuned LED model performs best on traditional metrics such as ROUGE, METEOR, MoverScore and BERTScore, while entity-based data filtering enhances specific factual consistency metrics. However, the JAENS approach underperformed, necessitating further investigation to understand its limitations. Additionally, the challenge of achieving high recall and precision in entity recognition was observed, which we plan to address in future work. We also intend to explore the reduction in traditional metric scores across different hallucination-mitigating designs, aiming to refine and optimize our approach for more reliable summary generation.

7

Automated Title Generation for Research Papers Using Pre-Trained and Large Language Models

In this chapter, we aim to generate titles of research papers from their abstracts using deep neural models. The title of a research paper communicates in a succinct style the main theme and, sometimes, the findings of the paper. Coming up with the right title is often an arduous task, and therefore, it would be beneficial to authors if title generation can be automated. In particular, we selected several pre-trained transformer models and fine-tuned them using a dataset of abstracts and titles. We believe that machine generated titles could be very useful for non-native English speakers who find it difficult to quickly construct a suitable title for their papers, as well as for novice researchers beginning their journey in the field. The generated titles can then be refined by them to more suit their own style and requirement.

7.1 Background and Motivation

This section provides an overview of the background related to title generation for research papers and the motivation behind this research.

7.1.1 Background

In academic publishing, the title of a research paper plays a crucial role in conveying the essence of the study. A well-crafted title not only informs potential readers about the core contributions of the paper but also serves as a key factor in attracting readership. The effectiveness of a title lies in its ability to encapsulate the main themes and findings in a concise and engaging manner. Previous research has shown that titles with certain characteristics, such as brevity and the inclusion of relevant keywords, tend to attract more citations and downloads [79, 96, 166]. Short and compact titles reportedly attract more citations [96]. Despite its importance, crafting an effective title remains challenging, particularly for non-native English speakers and novice researchers.

Automatic text summarization has a long history. As far back as 1958, Luhn et al. [112] pioneered an extractive summarization technique that selects sentences based on the frequency of significant words, excluding common words, to summarize technical papers and magazine articles. Lloret et al. [109] created a corpus of computer science papers from [arXiv.org](https://arxiv.org) with paired (Introduction, Abstract) sections.

Advancements in summarization techniques have followed with the development of sequence-to-sequence models [185], attention-based encoders with beam search and RNN [16, 129], and pointer-generator networks with coverage mechanisms [174], primarily applied to news datasets. The introduction of the transformer architecture [195] marked a significant shift, leading to the development of various pre-trained language models such as T5 [155], BART [50], and PEGASUS [222]. These models, initially trained in a self-supervised manner on broad, general-purpose text datasets, capture linguistic patterns and knowledge that can be fine-tuned for specific tasks across various domains.

In the context of title generation, researchers have explored methods to generate titles from their content. Tan et al. [187] proposed a coarse-to-fine approach that first identifies important sentences using hierarchical attention within an encoder-decoder framework. They tested their model on the *New York Times* (NYT) and DUC-2004 datasets. Our proposal acknowledges that a title essentially acts as a summary of the content within a research paper [134]. Mishra et al. [126] proposed a method for generating titles for academic papers by first creating a pool of candidate titles using a pre-trained model, selecting the most suitable one, and then refining it to ensure semantic and syntactic accuracy, thus enhancing the representativeness of the titles. They used datasets from [arXiv](https://arxiv.org) ¹, ACL [202], and ICMLA [193]. Recently, Liu et al. [106] used a pre-trained transformer encoder, OAG-BERT, on computer science research papers (including metadata) and used it to generate paper titles. These titles were found

¹<https://tinyurl.com/y9pu6xyp>

to be quite acceptable to researchers, although extensive evaluation using automatic measures was not conducted. In contrast, our work employs a diverse set of encoder-decoder and decoder-only transformers to generate titles from abstracts and includes a detailed evaluation of the generated titles. Our work also aligns with contemporary research [105, 40] exploring the potential of ChatGPT and other artificial intelligence tools in education, research, and scholarly publishing.

Given the tremendous success of neural language models, especially large language models (LLMs), in various natural language processing (NLP) tasks, it is natural to ask if titles of scientific papers can also be generated by these models. Given the expectation that a title should capture the key import of a paper and yet be a short sequence of words that appeal to a large readership, the task is a challenging one. Title generation can be considered as a special case of **abstractive text summarization** that aims to distil out the most crucial information from a document.

7.1.2 Motivation

The advent of large language models (LLMs) has revolutionized various natural language processing (NLP) tasks, leading to significant advancements in text generation, summarization, translation and, many more. Given the success of these models, it is natural to explore their potential in generating research paper titles. The motivation behind this research stems from the need to assist researchers, especially those who may struggle with language barriers or lack experience in academic writing. By automating the title generation process using pre-trained models fine-tuned on relevant datasets, we aim to provide a tool that can suggest appropriate titles, which researchers can then refine to match their style and specific requirements.

7.2 Challenges and Opportunities

This section discusses the current challenges in generating titles for research papers and explores potential advancements based on large language models or pre-trained language models.

7.2.1 Challenges

Generating an effective research paper title is inherently challenging due to the need for brevity, clarity, and informativeness. Titles must distill the most crucial aspects of a paper into a short phrase or sentence while still capturing the attention of the target audience. Additionally, the task of title generation can be seen as a special case of **abstractive text summarization**, where the goal is to condense the content of the

paper into a form that is not only representative but also engaging. The complexity of this task is compounded when using machine-generated titles, as ensuring that the generated titles are both accurate and contextually relevant requires sophisticated models and rigorous evaluation methods.

7.2.2 Opportunities

The use of PLMs, LLMs, and other advanced neural models in abstractive text summarization presents significant opportunities. These advancements can greatly enhance the generation of effective titles for research papers. Automating this process could streamline researchers' workflows, enabling them to concentrate more on their paper's content rather than the complexities of crafting titles. Moreover, this approach could democratize access to high-quality academic writing tools, particularly benefiting non-native English speakers and early-career researchers. The development and fine-tuning of models for this purpose also open avenues for further research into improving the quality and reliability of AI-generated academic content, thereby enhancing the broader field of NLP in scholarly publishing.

7.3 Main Contributions

The main contributions of this chapter are:

1. We fine-tuned pre-trained models, namely, **T5-base** [153], **BART-base** [98], and **PEGASUS-large** [223] with CSPubSum dataset to automatically generate titles of research papers. We have also used the open large language model **LLaMA-3-8B** consist of 8 billion parameters [189, 2]. We have contrasted its generation quality with and without fine-tuning it for this task. We have also used **ChatGPT 3.5** in a zero-shot setting to generate title from abstracts.
2. We have also curated a dataset of abstracts and titles, called **LREC-COLING-2024**, on which we evaluate our fine-tuned models without further training.
3. We evaluate the performance of the models using **ROUGE** [102], **METEOR** [17], **MoverScore** [227], **BERTScore** [224] and **SciBERTScore** metrics. We introduce **SciBERTScore** to measure the performance of the models as a variant of **BERTScore**. We also evaluate the factual accuracy of the model-generated titles at the entity level by employing metrics such as precision-source and F1-target, which were introduced in a study by [130]. Manual evaluation of a small subset of the generated titles has also been carried out.

4. A demo that allows the user to choose a pre-trained language model to generate a title from the abstract of a research paper is hosted at:
<https://title-generation-researchpapers.onrender.com/>. Additionally, on Hugging Face, we have publicly released the fine-tuned models² as well as the *LREC-COLING-2024* corpus³.

7.4 Methodology

In this section, we detail the fine-tuning process of various pre-trained language models (PLMs) and large language models (LLMs). We have chosen the following models:

1. **T5-base** [153]: It is an encoder-decoder model which is a slight variation of the original Transformer model [195]. Formulating every text processing problem, including translation, question answering, and classification, as a “text-to-text” transformation problem, the same model is trained to perform these diverse tasks. To pre-train the model, random text spans are corrupted/dropped and the model is trained to generate them. *T5-base* contains 220M parameters. We fine-tuned the model with the train subset of CSPubSum dataset.
2. **BART-base** [98]: A denoising autoencoder, it combines bidirectional and autoregressive transformers exemplified by BERT [50] and GPT [30], respectively. To train BART, the input text is first corrupted with a noising function, and then the model reconstructs the original text. It is particularly suitable for text generation problems. *BART-base* is configured with 139M parameters. Like *T5-base*, it is also fine-tuned on train set of CSPubSum dataset.
3. **PEGASUS-large** [223]: It is a Transformer-based encoder-decoder model that is trained on large text collections with an objective function specifically focused on summarization. In particular, pre-training the model involves masking *important* sentences from an input document and generating them as one output sequence. *PEGASUS-large* contains 568M parameters. Like the previous two models, we fine-tuned the *PEGASUS-large* model on training examples from the CSPubSum dataset.
4. **ChatGPT-3.5**: We use the instruct-tuned 7B version of the GPT-3 family model, identified as *gpt-3.5-turbo-1106*, commonly known as ChatGPT 3.5⁴. It is a decoder-only model that has been pre-trained on massive text corpora and then

²<https://huggingface.co/TRnlp>

³<https://huggingface.co/datasets/TRnlp/LREC-COLING-2024-Abstract-Title>

⁴<https://chatgpt.com/>

further trained with instruction fine-tuning and reinforcement learning with human feedback. It is a large language model with 175B parameters. We use the prompt-based in-context learning setup where we simply prompt the model to generate a title given the abstract.

5. **LLaMA-3-8B:** We use instruction-tuned *LLaMA-3-8B* model with fine-tuning and without fine-tuning. The LLaMA series of models [189] are decoder-only transformer-based large language models trained exclusively on publicly available datasets (in contrast to the ChatGPT and the GPT series). We fine-tuned the *LLaMA-3-8B model (with 8B parameters)* on the train subset of the *CSPubSum* dataset.

Note that although the above pre-trained models and LLaMA-3 come in multiple sizes, we choose the smallest sizes that are freely available due to constraints on our computational resources. Although the paper that proposed the PEGASUS model [223] discusses a base and a large model, we could only find the checkpoints for *PEGASUS-large* in Hugging Face⁵ and so, we consider this version. Although GPT-4 is more powerful than GPT-3.5, the former is only available to paid subscribers, and therefore, we do not use it for this research. However, we do expect that, like other deep learning applications, using larger models will enhance the quality of the generated titles.

7.5 Experimental Setup

In this section, we introduced a new dataset, discussed the datasets used, detailed the data pre-processing procedures, outlined the implementation specifics, and explored several automatic evaluation metrics.

7.5.1 Dataset Construction

We crawled an additional set of 1000 examples from accepted papers at LREC-COLING 2024⁶. This dataset, which we hereafter referred to as *LREC-COLING-2024*, comprises pairs of *abstract*, *title* for each paper.

7.5.2 Datasets Used

We have used the CSPubSum dataset provided by Collins et al. [43], which contains URLs of 10147 computer science publications from ScienceDirect⁷. We crawled the dataset and organized each example as a pair of *abstract* and *title*. The dataset is split

⁵<https://huggingface.co/google/pegasus-large>

⁶<https://aclanthology.org/events/coling-2024/>

⁷<https://www.sciencedirect.com/>

TABLE 7.1: Some statistics of CSPubSum dataset and LREC-COLING-2024 dataset. #Max: Number of maximum words; #Avg: Number of average words; #Min: Number of minimum words. Similarly used for sentences.

Split	Abstract		Title	
	Words	Sentences	Words	Sentences
	(#Min, #Max, #Avg)	(#Min, #Max, #Avg)	(#Min, #Max, #Avg)	(#Min, #Max, #Avg)
Train (8120)	23, 994, 185	1, 49, 8	2, 60, 11	1, 3, 1
Val (1014)	38, 1304, 179	2, 35, 8	2, 32, 13	1, 2, 1
Test (1013)	44, 1166, 194	2, 54, 8	3, 26, 12	1, 2, 1
LREC-COLING-2024 (Test) (1000)	58, 290, 162	1, 14, 7	3, 25, 10	1, 2, 1

into 8120 examples for training, 1014 examples for validation, and 1013 examples for testing.

We only train the selected deep neural language models on the train subset of CSPubSum dataset only but test them on the test subset of *CSPubSum* dataset and the corpus of *LREC-COLING-2024*. While *CSPubSum* dataset focuses on diverse topics of computer science, they primarily belong to the pre-transformer [195] era since the impact of transformers was produced after 2017 when CSpubSum was released. We wanted to verify how well the trained models perform on a different and more recent dataset on which it is not explicitly trained. So we curated *LREC-COLING-2024* dataset. The average number of tokens in a title in CSpubSum is 12 while it is 11 for LREC-COLING-2024, although a few titles are longer and some as small as two tokens. A few titles have multiple sentences; a title with two sentences is “Comparative statics effects independent of the utility function. When do we act the same way under risk?”. However, around 82% papers in *CSPubSum* and 90% papers in *LREC-COLING-2024* have no more than 15 tokens in their titles. So when fine-tuning or performing inference with the pre-trained models or large language models for title generation, we set the maximum title length to 20 tokens (recollect that due to sub-word generation, token count in a model generally exceeds the raw token count). Another reason for the imposed token limit is that shorter titles are generally more attractive to readers [96]. The dataset statistics for the two datasets used for title generation are presented in Table 7.1.

7.5.3 Data Processing

We removed only extra spaces from the documents in both datasets. We only retain examples where the abstract length is at least 20 tokens and the paper title length is at least 3 tokens. Since we impose a limit of 20 tokens on generated title length, we stipulate the abstract to be longer than 20 tokens so that the task can be treated as a text summarization problem.

7.5.4 Implementation Details

We have chosen the following pre-trained models from the Hugging Face repository for fine-tuning on the CS PubSum dataset: T5-base⁸, BART-base⁹, PEGASUS-large¹⁰. Fine-tuning was performed for 5 epochs. We used batch size of 32, learning rate of 4e-5, and the `metric_for_best_model` as ROUGE1-F1.

We use the following prompt to generate titles using LLMs, namely ChatGPT-3.5 and LLaMA-3-8B:

Create a concise title from this abstract using at most 20 tokens,
highlighting the main contributions and focus. <ABSTRACT>

Note that during evaluation, ChatGPT-3.5 and LLaMA-3-8B models were both given the above prompt without any in-context examples. Fine-tuning LLaMA-3-8B is an extremely computationally expensive task. Therefore, we use the parameter-efficient fine-tuning technique called Low-Rank Adaptation (LoRA). In particular, we have taken the model from Hugging Face, LLaMA-3-8B¹¹, then loaded it in 4-bit precision to save memory, and finally, fine-tuned it for 5 epochs using a learning rate of 4e-5, train batch size of 32 and eval batch size of 1, rank of the adaptation matrices($r=16$), `lora_alpha = 16` (Scaling Factor), and using `peft_config` for loading LoRA. For fine-tuning and evaluation after fine-tuning, we have used the same prompt that we mentioned above. For comparison, we also include results obtained with LLaMA-3-8B that is *not fine-tuned* on this dataset, i.e., it is used in zero-shot in-context learning without fine-tuning. We denote this variant as LLaMA-3-8B*. In case of ChatGPT-3.5, we use a temperature setting of 0.3, to generate titles based on the abstracts.

For all models, we have set the maximum number of input tokens (i.e., the abstract length) to 512 and output tokens (i.e., the title length) to 20. In all cases, we have measured the memory and compute power consumption using the WandB tool¹². The fine-tuning of the LLaMA-3-8B model took 1 hour 43 minutes for 5 epochs. In the context of pre-trained language models, PEGASUS-large model took 23 minutes, BART-base model took 7 minutes, and T5-base took 13 minutes for 5 epochs of fine-tuning. Training and fine-tuning of the models were done on Tesla A100-SXM4-40GB Colab Pro+ that supports GPU-based training.

⁸<https://huggingface.co/t5-base>

⁹<https://huggingface.co/facebook/bart-base>

¹⁰<https://huggingface.co/google/pegasus-large>

¹¹<https://huggingface.co/unsloth/llama-3-8b-bnb-4bit>

¹²<https://wandb.ai/site>

7.5.5 Evaluation Metrics

We used commonly used automatic text summarization evaluation metrics, including ROUGE [102], METEOR [17], MoverScore [227], BERTScore [224] and SciBERTScore [21], to assess the quality of generated title with the author-written title. ROUGE scores measure n -gram overlap between the generated and ground-truth titles. We use ROUGE-1, ROUGE-2, and ROUGE-L where the first uses unigram overlap, the second bigram overlap and the last compares the longest common subsequence between the generated title and the golden title. METEOR measures sentence-level accuracy based on the alignment between the generated text and the reference text. In contrast, MoverScore, BERTScore, and SciBERTScore aim to measure the *semantic similarity* between the output and the true title, using latent representations (embeddings) of the texts. MoverScore combines Word Mover’s Distance [93] and contextualized embeddings of the output and the ground-truth titles for semantic matching [227]. BERTScore calculates the cosine similarity between the BERT embeddings of the words in the two text sequences [224]. Since the current application is in the domain of computer science, we also proposed a variant of BERTScore where we use SciBERT [21] to generate the embeddings, then we compute their cosine similarity as in BERTScore; we call the modified metric, SciBERTScore.

These metrics are, however, inadequate to quantify *factual consistency* [90], which in our case connotes whether the entities present in the generated titles are precisely from the given abstract, and how much the entities in the generated title overlap with those in the author-written one. Hence, we have also used three new metrics, introduced by [130], to evaluate the factual consistency of the generated title. Note that we have utilized the same metrics in Chapter 6, Subsection 6.6.4, where we studied hallucinations in summary generation.

Let us call the author-written title as the target t , the model-generated title as the hypothesis h and the input abstract as the source s . We define $\mathcal{N}(t)$ as the count of named-entities in the target (author-written title) and $\mathcal{N}(h)$ as the count of named-entities in the hypothesis (model-generated title). $\mathcal{N}(h \cap s)$ is the number of entity matches across the generated title and the input source document (abstract). Some named-entities in the title span multiple words. If even some words of a named entity in the title occur in the abstract, we consider it as a successful entity match. We used the scispaCy model `en_core_sci_sm`¹³ to identify entities. **Precision-source**, defined as per the Equation 7.1:

$$prec_s = \mathcal{N}(h \cap s) / \mathcal{N}(h) \quad (7.1)$$

¹³<https://allenai.github.io/scispaCy/>

Precision-source ($prec_s$) is a metric that is used to determine the intensity of hallucination in relation to the input source document (abstract). Note that $prec_s$ represents the percentage of entities mentioned in the generated title that can be retrieved from the input source document (abstract). Low $prec_s$ indicates the possibility of hallucination in the generated title. However, $prec_s$ does not capture the generated title’s entity-level correctness in relation to the author-written title. Therefore, recollecting that the author-written title is the target t , we define target-level entity accuracy in terms of **precision-target** and **recall-target**, defined as per the Equations 7.2 and 7.3:

$$prec_t = \mathcal{N}(h \cap t) / \mathcal{N}(h); \quad (7.2)$$

$$recall_t = \mathcal{N}(h \cap t) / \mathcal{N}(t) \quad (7.3)$$

Finally, **F1-target** is calculated using Equation 7.4:

$$F1_t = \frac{2 * (recall_t * prec_t)}{recall_t + prec_t} \quad (7.4)$$

Here, $\mathcal{N}(h \cap t)$ represents the number of matched named-entities in the generated title and the author-written title.

Note that we have calculated the above mentioned *precision-source*, *precision-target*, *recall-target*, and *F1-target* in two ways. One is to consider the entity mentioned in each document (which may be the source s or target t or hypothesis h) as a set so that multiple occurrences of an entity in a document are equivalent to a single occurrence. The other is to consider the entity mentioned in a input source (abstract) as a list; here, if a metric is defined as $\mu = \mathcal{N}(x \cap y) / \mathcal{N}(x)$, then for each entity mention in x , we check if it occurs in y , and if so, increment the intersection count $\mathcal{N}(x \cap y)$ by unity; this approach is followed in [130]. In the first approach, we denote the metrics as $prec_s^U$, $prec_t^U$, $recall_t^U$, and $F1_t^U$ (U indicates that only unique entity mentions are considered). In the second, we represent them as $prec_s^{NU}$, $prec_t^{NU}$, $recall_t^{NU}$, and $F1_t^{NU}$ (NU denoting *not unique*).

7.6 Results

7.6.1 Quantitative Comparison of Various Fine-Tuned Models

In this sub-section, we report the results of experiments on the test dataset of *CSPubSum* dataset as well as *LREC-COLING-2024* dataset. Table 7.2 shows the performance in terms of ROUGE, METEOR, and semantic metrics like MoverScore, BERTScore and SciBERTScore, for the CSPubSum test dataset. Evaluation with entity-level factual

consistency metrics precision-source($prec_s^{NU}$, $prec_s^U$), precision-target ($prec_t^{NU}$, $prec_t^U$), recall-target ($recall_t^{NU}$, $recall_t^U$), and F1-target ($F1_t^{NU}$, $F1_t^U$) are shown in Table 7.3. The overall finding is that **PEGASUS-large** fine-tuned on the CSPubSum dataset achieves the highest scores for all the above metrics except precision-target metric. Table 7.4 and Table 7.5 show that the same **PEGASUS-large** model, that has been fine-tuned on the CSPubSum dataset, achieves the best performance on the LREC-COLING-2024 dataset except on BERTScore metric. Thus, the superlative performance of **PEGASUS-large** carries over to this dataset although it is not fine-tuned on it.

TABLE 7.2: Evaluation of all used models: F1-scores for ROUGE, METEOR are used for evaluating the both word-level matching and MoverScore, BERTScore, and SciBERTScore are used to compute the semantic fidelity of the generated titles for the CSPubSum dataset. All scores in percentage (%).

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore	SciBERTScore
T5-base	44.25	25.04	38.92	38.36	38.09	89.9	76.06
BART-base	45.7	25.97	40.11	39.37	39.75	90.21	76.89
PEGASUS-large	46.75	27.13	40.67	42.61	40.43	90.35	76.93
LLaMA-3-8B*	28.4	12.58	24.6	27.17	21.42	86.34	66.65
LLaMA-3-8B	40.8	21.23	36.57	34.5	37.02	89.99	76.41
ChatGPT-3.5	42.81	21.16	36.55	35.12	37.39	88.66	76.28

TABLE 7.3: Evaluation of all used models: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated title for CSPubSum dataset. All scores in percentage (%).

Model Name	$prec_s^{NU}$	$prec_s^U$	$prec_t^{NU}$	$recall_t^{NU}$	$F1_t^{NU}$	$prec_t^U$	$recall_t^U$	$F1_t^U$
T5-base	97.1	97.08	59.3	51.82	52.38	59.08	51.58	52.17
BART-base	97.44	97.39	61.52	52.84	53.91	61.35	52.56	53.72
PEGASUS-large	98.13	98.08	60.39	56.49	55.39	60.17	56.21	55.18
LLaMA-3-8B*	67.7	67.7	39.19	40.48	36.81	38.96	39.98	36.48
LLaMA-3-8B	78.95	78.93	57.54	46.99	49.12	57.44	46.97	49.08
ChatGPT-3.5	92.73	92.73	59.79	50.95	51.7	59.79	51.01	51.74

TABLE 7.4: Evaluation of all used models: F1-scores for ROUGE, METEOR, MoverScore, BERTScore and SciBERTScore for LREC-COLING-2024 dataset. All scores in percentage (%).

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore	SciBERTScore
T5-base	46.84	28.7	41.69	39.88	39.6	88.71	76.05
BART-base	46.87	27.66	41.89	38.93	39.61	88.84	75.94
PEGASUS-large	49.85	30.51	43.93	43.23	41.66	89.1	76.74
LLaMA-3-8B*	32.92	16.66	27.66	30.61	25.68	86.77	67.42
LLaMA-3-8B	45.3	26.53	40.51	38.18	38.93	88.83	76.14
ChatGPT-3.5	45.16	23.97	38.88	37.45	38.85	89.54	75.64

A deep dive into the quantitative performance: Now let us take a more nuanced view of the performance tables. Consider Table 7.2. It is immediately clear that the performance of *LLaMA-3-8B** (without fine-tuning) is considerably worse than that

TABLE 7.5: Evaluation of all used models: precision-source, precision-target, recall-target and F1-target in terms of $prec_s^{NU}$, $prec_s^U$, $prec_t^{NU}$, $prec_t^U$, $recall_t^{NU}$, $recall_t^U$, $F1_t^{NU}$ and $F1_t^U$ scores are used for evaluating the factual consistency of the generated title for LREC-COLING-2024 dataset. All scores in percentage (%).

Model Name	$prec_s^{NU}$	$prec_s^U$	$prec_t^{NU}$	$recall_t^{NU}$	$F1_t^{NU}$	$prec_t^U$	$recall_t^U$	$F1_t^U$
T5-base	97.44	97.39	63.06	59.24	57.17	62.8	58.88	56.87
BART-base	96.24	96.23	64.3	59.21	57.78	64.11	58.92	57.56
PEGASUS-large	97.87	97.85	66.32	64.64	61.47	66.15	64.30	61.28
LLaMA-3-8B*	74.6	74.52	45.29	48.16	43.48	45.02	47.64	43.13
LLaMA-3-8B	81.57	81.55	62.18	55.99	55.19	62.13	55.95	55.16
ChatGPT-3.5	91.98	91.98	60.87	57.66	55.56	60.84	57.69	55.56

of the other models which are fine-tuned on the domain-specific data. This suggests that fine-tuning is still useful to attain a good performance of the models. In contrast, *ChatGPT-3.5* is also used in zero-shot setup but it performs better, perhaps because it is constantly being used across the world and therefore, trained continually. A second observation is that although *LLaMA-3-8B* (fine-tuned) and *ChatGPT-3.5* have smaller ROUGE and METEOR scores compared to the smaller pre-trained models, their performance is closer to that of the latter according to the semantic metrics like SciBERTScore; for example, SciBERTScore of all models except *LLaMA-3-8B** lie between 76.06 and 76.93. This points to the highly abstractive nature of the LLMs output which has low word overlap with the original title yet displays impressive semantic similarity with it. Similar observations can be made from Table 7.4. Now let us look at Table 7.3. First note that the precision-source as well as F1-target do not vary much whether or not we count multiple occurrences of an entity. Again, *LLaMA-3-8B** (without fine-tuning) shows the lowest scores for these metrics. The smaller pre-trained models perform similarly. Consider precision source $prec_s^U$: its large value indicates that the entities in the generated summary are mostly present in the input source which is the abstract here. $prec_s^{NU} = 97.1$ for *T5-base*, 97.44 for *BART-base*, and 98.13 for *PEGASUS-large*. But $prec_s^{NU}$ is 92.73 for *ChatGPT-3.5* and 78.95 for *LLaMA-3-8B*, showing that these models generate novel words more frequently. In contrast, the F1-target ($F1_t^{NU}$) lies in the 50's for all models, meaning that there is only moderate between the entities in the ground-truth title and the generated title. In particular, $F1_t^{NU}$ for LLMs is lower than that for the smaller PLMs models, pointing to the abstractive nature of their output. We observe a similar pattern in Table 7.5 which reports the entity overlap statistics for the *LREC-COLING-2024* dataset. Empirically, we did not observe noticeable hallucination in the generated titles but we intend to undertake a detailed study of this aspect in a future work.

7.7 Case studies

We show three representative examples of title generation one from CSPubSum (test subset) and two from LREC-COLING-2024 to illustrate the behavior of the models. Figure 7.1 shows the example from CSPubSum, the fine-tuned *T5-base*, *BART-base*, and *PEGASUS-large* models have generated titles that have significant similarity with the author-written titles. However, *T5-base* appears to generate a very long title. *BART-base* and *PEGASUS-large* generated the same titles. In contrast, *LLaMA-3-8B* without fine-tuning, produced a title that resembled a one-sentence summary and even that is incomplete; it is more of an extract from the abstract. After fine-tuning, the *LLaMA-3-8B* model generated an acceptable title. *ChatGPT 3.5*, using prompt-based techniques, successfully generated a title that also captures the essence of the paper and is similar to other generated titles and the author-written title. However, despite its stylistic flair, the evaluation metrics (in Table 7.2) indicate that the titles generated by *ChatGPT-3.5* scored lower than those generated by the fine-tuned *PEGASUS-large* model. This discrepancy also highlights a limitation of the automated metrics in accurately evaluating titles generated in a highly abstractive and stylistic manner.

Author-written Title: “Comparative statics effects independent of the utility function. When do we act the same way under risk?”
fine-tuned T5-base: “Comparative statics effects independent of the utility function for portfolio choice and competitive firm under price uncertainty”
fine-tuned BART-base: “Comparative statics effects in the context of expected utility”
fine-tuned PEGASUS-large: “Comparative statics effects in the context of expected utility”
without fine-tuning LLaMA-3-8B: “The author proposes a methodological approach that enables comparative static analysis of various economic models, irrespective of the”
fine-tuned LLaMA-3-8B: “Comparative statics analysis: A new approach”
prompt based ChatGPT-3.5: “Comparative statics effects on expected utility in decision-making”

FIGURE 7.1: Input is an abstract from CSPubSum dataset. Titles generated by the different models are shown. Paper taken from <https://www.sciencedirect.com/science/article/abs/pii/S037722171500586X>.

Now let us look at Figure 7.2 which shows the outputs generated by the models for an example from *LREC-COLING-2024* dataset. In case of *BART-base*, the generated title is incomplete. This is due to the hard limit on the output token count. We have observed this problem with other models, too. A simple workaround (without retraining the models) is to increase the output token limit during inference, which

sometimes leads to a grammatically correct and complete title. While **T5-base** and fine-tuned **LLaMA-3-8B** generate a mostly correct title, they miss the “low-resource” setting which is correctly captured by **PEGASUS-large**. *ChatGPT-3.5* generates almost the same title as that of *PEGASUS-large* except that it uses the abbreviation “NMT”, and abbreviations may not be desirable in a title.

Author-written Title: “A Reinforcement Learning Approach to Improve Low-Resource Machine Translation Leveraging Domain Monolingual Data”
fine-tuned T5-base: “Reinforcement Learning Domain Adaptation for Neural Machine Translation”
fine-tuned BART-base: “A novel Reinforcement Learning Domain Adaptation method for Neural Machine Translation in the low-”
fine-tuned PEGASUS-large: “Reinforcement learning domain adaptation for Neural Machine Translation in the low-resource domain”
without fine-tuning LLaMA-3-8B: “Reinforced Domain Adaptation Method for Low Resource Neural Machine Translation”
fine-tuned LLaMA-3-8B: “Reinforcement learning domain adaptation for neural machine translation”
prompt based ChatGPT-3.5: “Reinforcement Learning Domain Adaptation for Low-Resource NMT”

FIGURE 7.2: Input is an abstract from LREC-COLING-2024 dataset. Titles generated by the different models are shown. Paper taken from <https://aclanthology.org/2024.lrec-main.132/>.

In the example in Figure 7.3, *PEGASUS-large* and *fine-tuned LLaMA-3-8B* have generated exactly the same title as the author-written one. A quick reading of the abstract of the input paper indicates that *T5-base* has captured an important aspect of the proposed model in the paper, namely the “entity abstraction approach”; however, the model has other characteristics and singling out one might not be appropriate. The title generated by *BART-base* is logically incomplete as it misses the part “for entailment tree generation”. *LLaMA-3-8B without fine-tuning* struggles to generate a succinct title. But ChatGPT-3.5 does generate an interesting title which captures the very purpose of the proposed model: “improving AI explanations”

Author-written Title: “A Logical Pattern Memory Pre-trained Model for Entailment Tree Generation”
fine-tuned T5-base: “An entity abstraction approach for logical pattern memory pre-trained models”
fine-tuned BART-base: “Logical pattern memory pre-trained model”
fine-tuned PEGASUS-large: “A logical pattern memory pre-trained model for entailment tree generation”
without fine-tuning LLaMA-3-8B: “The proposed method addresses the limitations of previous approaches by incorporating an external memory structure to capture the latent representations”
fine-tuned LLaMA-3-8B: “Logical pattern memory pre-trained model for entailment tree generation”
prompt based ChatGPT-3.5: “Improving AI Explanations with Logical Pattern Memory Pre-trained Model”

FIGURE 7.3: Input is an abstract from LREC-COLING-2024 dataset. Titles generated by the different models are shown. Paper taken from <https://aclanthology.org/2024.lrec-main.68/>.

7.8 Manual Evaluation

We selected 20 papers, ten from each dataset (note that we use the test subset in case of CSPubSum) for human evaluation. A human annotator working in the field of NLP was asked to choose the most appropriate title among the generated ones. It is found that for CSPubSum dataset, in 80% of the cases, the title generated by **PEGASUS-large** is most preferred, while for LREC-COLING-2024, in 50% cases, the output of **PEGASUS-large** wins while in 40% cases, the title generated by **ChatGPT-3.5** is considered the best.

7.9 Demo

A demo of the application is hosted at <https://title-generation-researchpapers.onrender.com/> where the user can input an abstract into a text box, select a suitable fine-tuned language model, and the maximum token count for the title, and obtain a title generated by the model. We have found that if the generated title is incomplete, increasing the token count (to say, 25 or 30) generally produces a correct and complete title.

The screenshot shows a web browser window with the address bar displaying 'title-generation-researchpapers.onrender.com'. The page title is 'Research Paper Title Generation'. The interface is divided into two main sections: 'Paper Contents' and 'Generated Title'.

Paper Contents: A text area containing the following text:

Common Design Structure Discovery (CDS) is to identify local structures shared by multiple models. Nowadays it is mainly restricted to part models. Extending it to assembly models can produce a significant value for assembly design reuse. However, current descriptions of assembly models usually capture topological information qualitatively, considering little geometric information, and thus are not suitable for CDS in assembly models (CDSDA). To counter this problem, this paper proposes a generic face adjacency graph (GFAG) which is extended from the face adjacency graph for B-Rep part model description. GFAG can transform abstract relationships in assembly models into measurable entities by introducing a concept of mating face pair (MFP), thus facilitating a more quantitative and consistent description of parts and relationships in assembly models. Corresponding to geometric faces and edges in a part model, GFAG treats parts

Generated Title: A text area displaying the generated title:

Generic face adjacency graph for common design structure discovery in assembly models

Form Elements:

- Select Preferred Model:** A dropdown menu with 'pegasus-large-Abstract-Title-CSPubSum' selected.
- Maximum Tokens:** A text input field containing the value '20'.
- Generate Title:** A green button labeled 'Generate Title'.

FIGURE 7.4: Graphical user interface of our pre-trained language model-based title generation application.

7.10 Discussion

In this study, we examined the performance of multiple pre-trained language models (PLMs) and large language models (LLMs) in generating research paper titles. Our results indicate that the fine-tuned **PEGASUS-large** model outperformed the other models on both the *CSPubSum* and *LREC-COLING-2024* datasets. This suggests that PEGASUS-large is particularly effective for generating research paper titles, benefiting from fine-tuning on domain-specific data.

The superior performance of PEGASUS-large underscores the importance of fine-tuning pre-trained models to specific tasks. It shows that even though larger models are computationally intensive, they can be highly effective for targeted tasks when fine-tuned properly. The high performance on the *LREC-COLING-2024* dataset, achieved without additional fine-tuning, indicates that pre-trained models have robust generalization capabilities.

Despite the high quality of the generated titles, there are limitations. The titles may not always fully capture the content or tone of the research papers, suggesting that further refinement by authors might be necessary. Additionally, while these tools are advantageous for non-native English speakers and novice researchers, they are not without limitations in ensuring perfect alignment with specific paper contexts.

7.11 Conclusion

This study demonstrated that pre-trained and large language models can effectively generate titles for research papers, with the fine-tuned PEGASUS-large model achieving the best results. This success highlights the potential of pre-trained models, particularly when fine-tuned on domain-specific datasets, to produce high-quality titles.

Overall, while the generated titles are of high quality, further refinement by authors is advised to better align with the content of their papers. These tools are especially helpful for non-native English speakers and early-career researchers, as they streamline the process of generating effective titles.

Future work should focus on analyzing discrepancies between author-assigned and AI-generated titles. This analysis will provide insights into the semantic and contextual differences between human and AI-generated titles, highlighting areas for improvement in model performance. Additionally, exploring methods to assess the quality of abstractive outputs from large language models (LLMs) will be essential for advancing automated title generation.

8

SilverCSPicoSum: A Dataset of Very Short Summaries Generated with ChatGPT-3.5

Researchers routinely encounter the challenging task of understanding the latest scientific progress or tools from vast repositories of scientific papers. The summary of a paper comes as a quick aid for such tasks. The tremendous advancements in Natural Language Processing, in particular the rise of large language models (LLMs), in recent years have made possible the automatic generation of high quality summaries of research papers. Recent research indicates that human annotators prefer summaries produced by large language models (LLMs) over the original reference summaries in widely used summarization datasets. In this chapter, we contributed a dataset *SilverCSPicoSum* that consists of abstracts of papers and their very short summaries generated by a LLMs, more specifically GPT-3.5. We evaluate a subset of this dataset with the help of human annotators. We also train smaller neural models on this dataset along with used some pre-trained models for fine-tuning. In this chapter we dealt with a novel task of generating *silver standard pico summaries* by prompting of *GPT-3.5-turbo-1106* model. Silver standard datasets, created using LLMs, offer a cost-effective and scalable solution for generating large training datasets, facilitating initial benchmarking and bridging the gap between manual and automated approaches.

8.1 Introduction

Recent advances in text summarization have focused on fine-tuning pre-trained models on domain-specific datasets [98, 223, 153, 152]. Large language models (LLMs) such as GPT-3 [30], GPT-4 [1], T0 [197], and PaLM [39] have demonstrated their flexibility and efficiency in adapting to various tasks with minimal fine-tuning. Although early automatic summaries faced issues like grammatical errors and hallucinations [228, 57], recent improvements have shown that LLM-generated summaries are often preferred over human-written ones [146]. Silver standard datasets, created using LLMs, offer a cost-effective and scalable solution for generating large training datasets, facilitating initial benchmarking and bridging the gap between manual and automated approaches.

The exponential growth of scientific literature [27] poses a challenge for researchers to stay updated, as the number of papers doubles approximately every nine years [194]. To address this, there is a growing trend to provide concise summaries or highlights in addition to abstracts [163, 165, 162, 164], which are easier to read and understand on handheld devices. Text summarization, which condenses documents to provide key information, can be approached through extractive [92] or abstractive methods [53]. While extractive methods select key sentences from the text, abstractive methods generate summaries by rephrasing the text with new words, often leveraging advanced text generation capabilities of LLMs for superior results.

This chapter contributed *SilverCSPicoSum* dataset, which is constructed using LLMs to provide a cost-effective and scalable resource for automatic scientific paper summarization. We aim to extract key insights from research papers using pre-trained and large language models to advance the field of text summarization.

8.2 Background & Related Work and Motivation

This section provides an overview of the background related to silver stand pico summary generation and explains the motivation behind our proposed dataset.

8.2.1 Background & Related Work

We present an background and the overview of the related literature on text summarization, focusing on the datasets and models used for scientific paper summarization.

In recent years, deep learning models have significantly impacted various fields, including natural language processing, computer vision, and scientific research. A major challenge for these models is the need for large, high-quality datasets to train and evaluate them effectively. For text summarization tasks, creating these datasets typically involves extensive manual work. Human annotators must read, analyze, and summarize

large volumes of text, which is both time-consuming and costly. As a result, manually created datasets are often limited in size, which can restrict the performance of deep learning models in practical applications. The shortage of large and diverse datasets hampers these models' ability to generalize across different domains. For text summarization tasks, we can address this challenge by leveraging the capabilities of large language models (LLMs) to produce high-quality datasets more quickly and affordably.

Datasets: High-quality domain-specific datasets are pivotal to the training of deep neural models for summarization, and therefore, the research community has invested significant efforts to create them. In the news domain, the most widely used datasets include CNN/Daily Mail, DUC, NY Times, where abstracts or highlights are utilized as reference summaries for model training. In the landscape of scientific paper summarization, CSPubSum is a pivotal dataset for extractive summarization, containing URLs of approximately 10K papers provided by Collins et al. [43]. ArXiv and PubMed are significant datasets for abstract generation from full papers, as detailed by Cohan et al. [41]. The SciSummNet dataset includes scientific paper abstracts with citation contexts and human-annotated summaries as ground truth, emphasizing the substantial and costly human effort required for producing high-quality summaries [214]. The TalkSumm dataset uses conference videos to automatically generate summaries, with human validation confirming their quality as comparable to manually created summaries [97]. *MixSub*, proposed by Rehman et al. [164], contains abstracts and author-written research highlights in various subject domains. SciTLDR is a multi-target dataset with 5,411 TLDRs for 3,229 computer science papers, featuring both author-written and expert-derived summaries [32]. Atri et al. [14] proposed the mTLDR dataset, which consists of 4,182 instances from conferences like ICLR, ACL, and CVPR, integrating videos, audio, text, and both author and expert summaries to support multimodal extreme abstractive summarization.

Models: The literature on scientific paper summarization techniques is extensive, beginning with Luhn et al. [112], which proposed an extractive approach based on word frequency and sentence position. Subsequent approaches, such as TF-IDF and LexRank, introduced statistical and graph-based methods for sentence selection [54]. Notable advancements include knowledge-based systems like FRUMP and SUMMONS, which use information extraction techniques. More recent methods leverage machine learning, with Kupiec et al. [92] introducing a Naive Bayes Classifier for scientific articles and Radev et al. [150, 149] proposing MEAD for multilingual, multi-document summarization. Subsequent advances, such as sequence-to-sequence models by Sutskever et al. [185] and attention mechanisms by Bahdanau et al. [16], greatly improved abstractive summarization. See et al. [174] developed pointer-generator networks to handle out-of-vocabulary words and repetition, with further enhancements using pre-trained

embeddings for better semantic representation [11, 163, 162]. Recent transformer-based models, including T5, BART, ProphetNet, and PEGASUS, have advanced NLP significantly, with LLMs like ChatGPT receiving high evaluations [64, 146]. Liu et al. [108] found that smaller models can match LLMs in automated assessments but not in human evaluations, while Zhang et al. [220] introduced SummIt, which uses LLMs for iterative summary improvement through self-evaluation and feedback.

We chose scientific papers as our domain of interest because they are densely loaded with technical information that can be difficult to quickly grasp. A concise overview would be highly valuable for readers with limited time, encouraging them to explore the full paper if they find the summary both relevant and insightful. However, manually annotating a large dataset requires many domain experts, which is very challenging and costly. Hence, in this chapter, we utilize the capability of large and pre-trained language models to generate extremely short summaries from the abstracts of papers. We restrict to abstracts, instead of the full text, because for in most cases abstracts are freely available while the full text resides behind a paywall. The abstract and pico summary pair can be used to train more complex and data-demanding summarization models.

8.2.2 Motivation

The motivation for our work arises from the significant time and cost associated with creating large-scale datasets for training deep learning models. While manually curated datasets are well known for their high accuracy, producing them at the scale needed for modern, data-hungry models is impractical. This issue is particularly urgent in the fields where deep learning depends on vast amounts of annotated data, such as summarization, where each document must be meticulously reviewed by experts.

To address this challenge, we propose the creation of a silver-standard dataset using *ChatGPT-3.5*. By leveraging large language models, we can generate high-quality summaries much more quickly and affordably than through manual methods. Although silver-standard datasets may not match the precision of human-generated summaries, they offer a scalable solution that facilitates the development and fine-tuning of deep learning models across various domains. This approach balances efficiency with quality, providing a valuable resource for the research community.

Our silver-standard dataset *SilverCSPicoSum* not only alleviates the burden of manual dataset creation but also supports the advancement of deep learning models by offering a large-scale resource that is otherwise challenging to obtain through traditional means.

8.3 Challenges and Opportunities

This section addresses the challenges associated with current datasets used for generating pico summaries and explores the opportunities provided by the newly introduced dataset.

8.3.1 Challenges

Existing datasets for identifying key contributions or research highlights of scientific papers are predominantly created manually. This approach is both time-consuming and expensive, as it requires extensive human effort to read, analyze, and summarize each paper. Additionally, the manual process limits the size, scalability, and diversity of these datasets, which can constrain the performance and generalization of deep learning models. The challenges associated with manual dataset creation hinder the development of more effective and efficient summarization tools.

8.3.2 Opportunities

Our contributed dataset addresses these challenges by providing machine-generated pico summaries derived from research paper abstracts. As a silver-standard dataset *SilverCSPicoSum*, it offers a scalable and cost-effective alternative to manual creation. Leveraging large language models, our dataset allows for the efficient creation of high-quality summaries, facilitating the training and evaluation of deep learning models. This approach not only reduces the time and cost associated with dataset creation but also enhances the availability of diverse, high-quality training data for summarization tasks.

8.4 Main Contributions

The main contributions of this chapter are:

1. **SilverCSPicoSum Dataset:** We introduced the *SilverCSPicoSum* dataset, a corpus of extremely short summaries created from ScienceDirect papers using GPT-3.5. This dataset reduces the cost and time needed for human-annotated datasets. We have adapted the Precision-source metric [130] to evaluate factual consistency in *SilverCSPicoSum*, showing high precision and minimal hallucination in the summaries. Additionally, human evaluations of the pico summaries confirmed their quality, scoring well in adequacy, fluency, coherence, and correctness.
2. **Model Fine-Tuning:** We have fine-tuned several pre-trained models, including T5 [153], BART [98], Pegasus[223], ProphetNet [148] on the *SilverCSPicoSum* dataset.

Additionally, we fine-tuned *LLaMA-2 7B & 8B, resp.* with LoRA on this dataset. Finally, we used a pointer-generator network with coverage mechanism [174, 164], that has earlier demonstrated high performance on several summarization tasks.

3. Evaluation: We have used ROUGE, METEOR, MoverScore, and BERTScore to evaluate model performance and conducted human evaluations to validate summary quality.

8.5 Dataset construction

CSPubSum was created from computer science papers. We initially crawled the dataset from ScienceDirect¹, incorporating URLs introduced by Collins et al. [43]. In the dataset construction process, we utilized the instruction-tuned variant of the GPT-3.5 family model, specifically *gpt-3.5-turbo-1106*, to generate pico summaries from the provided abstracts. The initial instruction for pico summary generation was: “*Can you generate a summary of the given text within 30 words?*”. During the first round of prompting, we found that many generated summaries exceeded the 30-word limit. In particular, we noticed that 138 of the generated pico summaries were even longer than 50 words. To address this, we refined our instruction to: “*Generate a summary of the given text within 30 words; the word count must not exceed 50. Adhere strictly to this limit*”. Despite this instruction, 37 summaries still exceeded the 50-word limit in the second round. Ultimately, we retained 10,137 summaries that were no longer than 50 words, and discarded the remaining examples. Table 8.1 provides a comprehensive comparison of *SilverCSPicoSum* with other prominent datasets in the scientific paper summarization field. This table provides key statistics including the number of documents, average word counts for inputs and summaries, data sources, summary origins, and the relevant domains of study. Figure 8.1 illustrates the statistical distribution

TABLE 8.1: Comparison of SilverCSPicoSum with existing datasets in the scientific paper summarization domain. #Documents represents number of documents, the average word counts for input & summary, the input data source, the summary origin, and the domain. Abstract, Introduction, and Conclusion are abbreviated as Abs, Int, and Con.

Dataset	#Documents	Average Words Input / Summary	Input Based On	Summary Source	Domain
CSPubSum	10.3K	8.2K / 226	Full papers	Author-written highlights	Computer Science
PubMed	133K	3K / 203	Full papers	Abstract	Biomedical (PubMed repo)
ArXiv	215K	4.9K / 220	Full papers	Abstract	ArXiv repo
SciSummNet	1.0K	4.7K / 150	Abstract + Citation Context	Human-annotated Summary	ACL Anthology Network
TalkSumm	1.7K	4.8K / 965	Full Paper & Video Talk	Alignment (Introduction & Transcript)	ACL, NAACL, EMNLP
MixSub	19.8K	148 / 57	Abstract	Author-written Highlights	Multi-Disciplinary ScienceDirect.com
SciTLDR	3.2K	5K / 21	Full Text (or Abs, Int, Con)	Author-written and/or Expert-Derived TLDR	Computer Science
SilverCSPicoSum (ours)	10.1K	184 / 32	Abstract	GPT-3.5 Generated Pico Summary	Computer Science

¹<https://www.sciencedirect.com/>

of abstract and pico summary lengths across the train, validation, and test datasets within the SilverCSPicoSum dataset, highlighting the mean, minimum, maximum, and standard deviation values for each category.

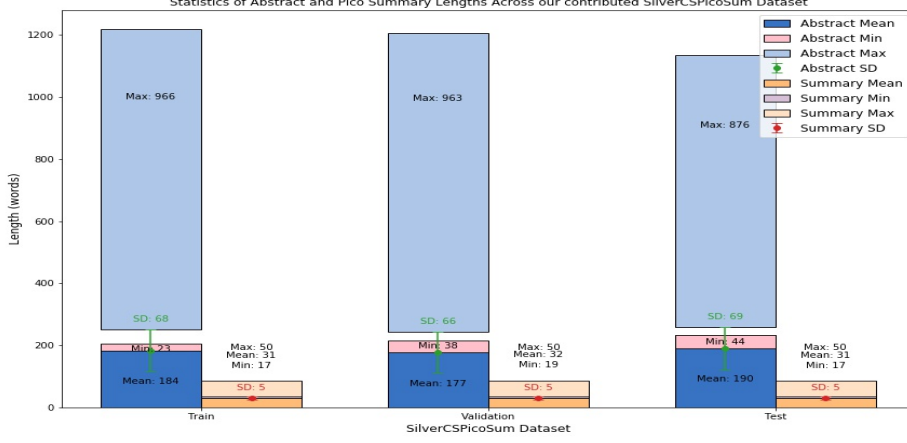


FIGURE 8.1: Length statistics mean, min, max, and standard deviation (SD) for abstracts and pico summaries in the SilverCSPicoSum dataset, across Train, Validation, and Test sets.

8.6 Quality Assessment for SilverCSPicoSum

8.6.1 Factual Consistency

To assess the factual consistency of our contributed *SilverCSPicoSum* dataset, we adapted the *Precision-source* metric from [130]. We define $\mathcal{N}(s)$ as the number of named entities in the abstract and $\mathcal{N}(t)$ as the number of named entities in the ChatGPT-3.5-generated pico summary. To determine the number of entities in the hypothesis that have corresponding matches in the source document, we use $\mathcal{N}(h \cap s)$. In cases where a named entity in the summary consists of multiple words, we treat it as a match if any part of the entity is found in the original document, allowing for partial matches using n -grams. We evaluated entity matching using two methods: treating entities as a set (counting unique occurrences) and as a list (counting all occurrences). Named entities were identified using the scispaCy model `en_core_sci_sm`². A low $prec_s$ indicates potential hallucinations. We use two approaches to calculate Precision-source ($prec_s$):

$$prec_s^U = \mathcal{N}(t \cap s)^U / \mathcal{N}(t)^U \quad (8.1)$$

$$prec_s^{NU} = \mathcal{N}(t \cap s)^{NU} / \mathcal{N}(t)^{NU} \quad (8.2)$$

²<https://allenai.github.io/scispaCy/>

where $\mathcal{N}(t \cap s)^U$ and $\mathcal{N}(t \cap s)^{NU}$ represent the number of matched entities in the summary and abstract for unique and all mentions, respectively. $\mathcal{N}(t)^U$ and $\mathcal{N}(t)^{NU}$ denote the total number of entities in the pico summary for unique and all mentions, respectively. Table 8.2 illustrates the factual consistency of the *SilverCSPicoSum* dataset. High $prec_s^U$ and $prec_s^{NU}$ scores, highlighted in blue, indicate minimal hallucination in the pico summaries.

TABLE 8.2: Detailed breakdown of Named-Entity counts and Precision Scores in the SilverCSPicoSum dataset across different splits.

	Non-Unique				Unique			
	Avg. $\mathcal{N}(s)^{NU}$	Avg. $\mathcal{N}(t)^{NU}$	Avg. $\mathcal{N}(t \cap s)^{NU}$	$prec_s^{NU}$	Avg. $\mathcal{N}(s)^U$	Avg. $\mathcal{N}(t)^U$	Avg. $\mathcal{N}(t \cap s)^U$	$prec_s^U$
Train (8111)	52.42	10.96	9.92	90.56	41.74	10.64	9.57	90.06
Val (1013)	52.49	11.66	10.55	90.74	41.83	11.24	10.10	90.14
Test (1013)	54.72	11.02	10.00	90.78	43.35	10.67	9.63	90.24

8.6.2 Human annotation of SilverCSPicoSum

Inspired by Friedrich et al. [58] on human evaluation of summaries, we randomly selected 10 abstract and pico summary pairs generated by ChatGPT 3.5 from our SilverCSPicoSum dataset. Three annotators evaluated the summaries on: (a) **Adequacy**: How much important information is captured? (‘1: Very little’, ‘2: A moderate amount’, ‘3: Most’); (b) **Fluency**: How fluent is the summary? (‘1: Dis-fluent English’, ‘2: Good English’, ‘3: Flawless English’); (c) **Coherence**: How coherent is the summary? (‘1: Not coherent’, ‘2: Moderately coherent’, ‘3: Highly coherent’); (d) **Correctness**: How accurate is the information? (‘1: Mostly incorrect’, ‘2: Mostly correct’, ‘3: Completely correct’). The average scores for adequacy, fluency, coherence, and correctness are 2.17, 2.44, 2.47, and 2.47, respectively, all of which are above moderate, indicating good summary quality.

8.7 Methodology

In this section, we discussed pre-trained language models, large language models (LLMs), and hybrid models used for summarization with the *SilverCSPicoSum* dataset. We explored different Hugging Face models, including T5 (small, base, large), BART (base, large), ProphetNet (large), and Pegasus (large). Additionally, we experimented with instruction-based fine-tuning on LLaMA-2-7B and LLaMA-3-8B, and employed a pointer-generator network with a coverage mechanism and SciBERT embeddings.

Pointer-generator + Coverage mechanism with SciBERT [164] combines the pointer-generator model [174] with the coverage mechanism [191], utilizing SciBERT embeddings [21]. Unlike the original pointer-generator model, which learns word embeddings from scratch, this approach uses pre-trained SciBERT to generate contextual embeddings based on input tokens.

T5 (Text-to-Text Transfer Transformer) [153] utilizes a transformer architecture [195] with an encoder-decoder setup. The encoder uses self-attention and feed-forward layers, while the decoder includes additional attention mechanisms to focus on encoder outputs. T5 supports various tasks, including abstractive text summarization, and can be fine-tuned with task-specific prefixes like “*summarize:*”. Variants of T5, including small, base, and large, offer different trade-offs between computational efficiency and performance.

BART [98], built on the transformer architecture [195], is a sequence-to-sequence model with a GPT-style auto-regressive decoder. It replaces ReLU activations with GeLUs and uses techniques like token masking, text infilling, and sentence permutation to enhance language understanding. BART excels in text generation, summarization, and comprehension tasks. Variants include BART base with 139 million parameters and BART large with 406 million parameters.

ProphetNet [148], built on the transformer framework [195], introduces an n -gram prediction objective, which forecasts the next n tokens rather than just one. This approach improves fluency and coherence in text generation. It features approximately 356 million parameters with 12 layers each in the encoder and decoder.

Pegasus [223] employs a transformer-based encoder-decoder framework with innovative self-supervised objectives, including CSG (Gap Sentences Generation) and MLM (Masked Language Model). CSG masks entire sentences to create a pseudo-summary, while MLM masks 15% of tokens in the input text. Pegasus large has 568 million parameters with 16 layers each in the encoder and decoder.

LLaMA 2 is a series of large language models, including LLaMA 2 and the dialogue-optimized LLaMA 2-Chat, with up to 70 billion parameters [190]. Built on the transformer framework [195], LLaMA 2 uses rotary positional embeddings (RoPE) [183], SwiGLU activation [176], and RMSNorm [219]. It features a 4,000-token context length and was trained on 2 trillion tokens. LLaMA 2-Chat, fine-tuned for dialogue, is available in 7B, 13B, and 70B configurations and incorporates advanced techniques like instruction tuning and Reinforcement Learning from Human Feedback (RLHF).

LLaMA 3 is an advanced language model with a decoder-only architecture and auto-regression [2]. Building on LLaMA 2, it features a refined 128k tokenizer and supports 8 billion and 70 billion parameters. It uses grouped query attention (GQA) and Rotary Positional Encoding (RoPE) for enhanced performance. Trained on 15 trillion tokens,

LLaMA 3 handles sequences up to 8192 tokens and is available in both pre-trained and instruction-tuned versions. Figure 8.2 outlines the framework, including the *SilverC-SPicoSum* dataset construction, model application, and evaluation metrics.

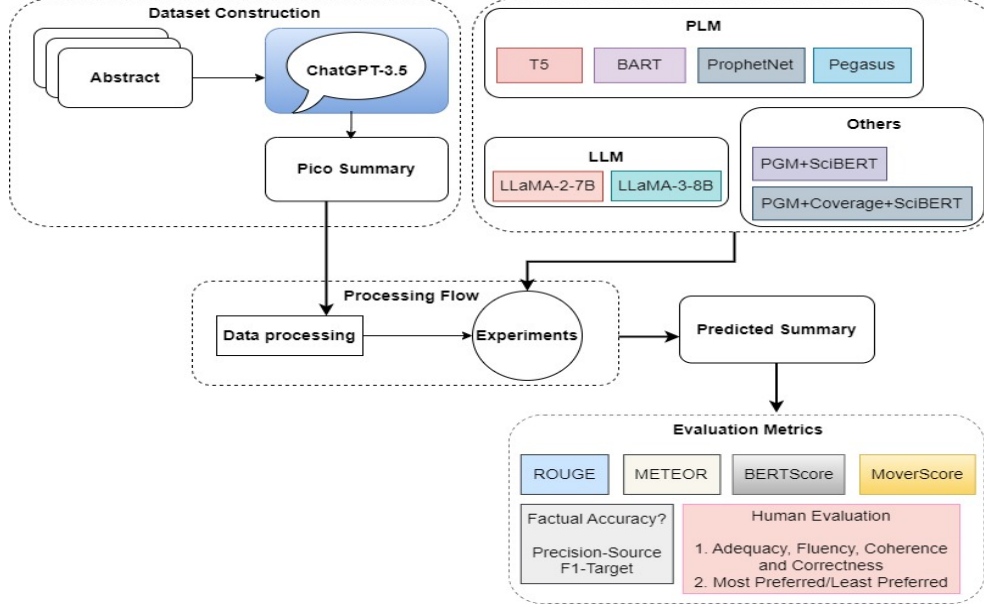


FIGURE 8.2: Framework showing the SilverCSPicoSum dataset construction, model application, and evaluation metrics.

8.8 Experimental setup

In this section, we have discussed some basic pre-processing steps, and implementation details.

8.8.1 Dataset pre-processing

We performed basic pre-processing by removing whitespace and filtered out examples with abstracts shorter than 20 tokens and pico summaries shorter than 10 tokens.

8.8.2 Implementation details

We trained a pointer-generator network with a coverage mechanism and SciBERT embeddings ($PGM + SciBERT + Coverage$), initially using pre-trained SciBERT embeddings, then embedding are fine-tuned them during training. The model was trained up to 11 epochs (11,252 steps) with SciBERT embeddings, followed by an additional 1.5 epochs (1,575 steps) with the coverage mechanism. Both models used 768-dimensional embeddings, mini-batches of size 8, bidirectional LSTMs with a cell size of 256, and

a vocabulary size of 50,000 tokens. The learning rate was 0.15 with gradient clipping at a maximum norm of 1.2, and `SciVocab-uncased`³ was used for tokenization. We used other hyperparameters as suggested by [174]. We have used the validation set to determine the number of epochs for training.

For fine-tuning *T5* [153], *BART* [98], *ProphetNet* [148], and *Pegasus* [223], we used a batch size of 8 and fine-tuned for up to 10 epochs with a learning rate of 4e-5, an evaluation strategy of epoch, and ROUGE1-F1 for metric selection. Pre-trained models from Hugging Face included *T5* (small, base, large), *BART* (base, large), *ProphetNet-large*, and *Pegasus-large*.

We fine-tuned *LLaMA-2-7B*⁴ with Low-Rank Adaptation (LoRA) in 8-bit precision for 5 epochs. We used a learning rate of 4e-5, train and eval batch size of 8, `r=8` (Rank of the Adaptation Matrices), `lora_alpha` (Scaling Factor)=8, set `lora_dropout=0.05`, set `load_best_model_at_end` to `true`, and used `evaluation_strategy` as epoch. LoRA adapts pre-trained models to specific tasks with lower computational costs and memory requirements by learning low-rank updates to the model weights. Similarly, we fine-tuned *LLaMA-3-8B*⁵ in 4-bit precision with the same settings.

All models were fine-tuned on Tesla A100-SXM4-40GB GPUs via Colab Pro+, with a maximum input of 512 tokens and target summaries limited to 60 tokens.

To evaluate model performance, we used ROUGE [102], METEOR [17], MoverScore [227], and BERTScore [224].

8.9 Results and Analysis

In this section, we present the evaluation results for various models on the *SilverCSPicoSum* dataset using abstracts as input. To evaluate model performance, we used ROUGE [102], METEOR [17], MoverScore [227], and BERTScore [224] as described Chapter 2, Section 2.7. Table 8.3 shows the F1-scores for ROUGE-1, ROUGE-2, ROUGE-L, METEOR, MoverScore, and BERTScore, with all scores expressed as percentages. The fine-tuned **LLaMA-3-8B** model outperforms others across all metrics, achieving the highest F1-scores for ROUGE, METEOR, MoverScore, and BERTScore. Table 8.3, the last column shows the total parameters used for executing a deep learning model or fine-tuned PLMs or LLMs. Although *LLaMA-3-8B* and *LLaMA-2-7B* have 8 billion and 7 billion parameters, respectively, only a fraction is utilized.

³https://huggingface.co/allenai/scibert_scivocab_uncased/

⁴<https://huggingface.co/meta-llama/Llama-2-7b-hf>

⁵<https://huggingface.co/unsloth/llama-3-8b-bnb-4bit>

TABLE 8.3: Evaluation of all models: F1-scores for ROUGE, METEOR, MoverScore, and BERTScore on the SilverCSPicoSum dataset using abstracts as input. All scores are presented as percentages (%).

Model Name	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	MoverScore	BERTScore	Parameters
PGM + SciBERT	43.97	22.76	36.34	41.60	32.16	89.10	21.5M
PGM + Coverage + SciBERT	44.75	22.06	36.01	41.82	33.32	89.10	21.5M
T5-small	52.51	31.84	45.07	51.33	42.01	91.60	60M
T5-base	57.39	36.68	49.99	56.62	47.41	92.61	220M
T5-large	40.05	19.35	31.45	38.12	28.65	89.08	737M
BART-base	54.87	33.96	47.63	53.95	45.57	92.31	139M
BART-large	57.04	35.96	49.45	57.39	47.34	92.66	406M
ProphetNet-large	55.96	34.63	48.74	51.98	47.10	91.57	391M
Pegasus-large	55.88	34.91	48.42	54.81	46.04	92.41	568M
LLaMA-2-7B	48.70	26.92	41.51	48.55	41.54	91.55	8.39M
LLaMA-3-8B	59.50	38.99	52.44	58.15	50.09	93.18	6.82M

8.9.1 Manual evaluation

Prior to manual evaluation, we assessed 50 examples from each model’s predicted pico summaries using ChatGPT-3.5 (gpt-3.5-turbo-1106). We employed the prompt ‘Given two paragraphs, P1: “ $\langle abstract \rangle$ ” and P2: “ $\langle predicted_summary \rangle$ ”, determine if P2 contains any hallucination relative to P1. Answer either “Yes” or “No”. Note that $\langle abstract \rangle$ and $\langle predicted_summary \rangle$ are placeholders for the abstract and the predicted summary, respectively. This prompt was executed three times, and the majority response was used to identify any hallucinations. Remarkably, none of the 50 examples from any model exhibited hallucinations.

We manually evaluated model generated pico summary with respect to the silver standard pico summary by two way.

First method: For human evaluation, we prepared four sets, each consisting of 10 papers with their abstracts and one of the following: (1) silver standard pico summaries (**A**), (2) summaries generated by the fine-tuned *T5-base* model (**M1**), (3) summaries generated by the fine-tuned *BART-large* model (**M2**), and (4) summaries generated by the fine-tuned *LLaMA-3-8b* model (**M3**), unknown to the human annotators. We selected the summaries from these models because their automatic metric scores, as shown in Table 8.3, outperformed those of other models across various evaluation metrics. Each set was evaluated by three raters, who assessed the summaries based on four parameters: **Adequacy**, **Fluency**, **Coherence**, and **Correctness**, as previously described in subsection 8.6.2. All human annotators possess or are pursuing advanced degrees in software engineering at premier universities in India. Table 8.4 shows that summaries generated by the fine-tuned *LLaMA-3-8B* model received higher average ratings for the parameter of **correctness** compared to other models.

Second method: Additionally, we asked annotators to select the most and least

preferable summaries from the provided options. We used 10 papers and their abstracts along with the silver standard pico summary as ground truth. We also provided summaries generated by four different models to 15 annotators. The models included fine-tuned versions of *T5-base*, *BART-large*, *LLaMA-3-8B*, and (*PGM + SciBERT + Coverage*). We chose three fine-tuned pre-trained large language models along with a neural abstractive encoder-decoder model to compare the quality of their generated summaries. They could choose up to two summaries or none at all in each category. The total number of times each model-generated summary was chosen as the most or least preferable is reported in Table 8.5.

TABLE 8.4: Human evaluation score of the summary generated by the 3 fine-tuned models with respect to the silver standard pico summary from SilverCSPicoSum dataset, based on Adequacy, Fluency, Coherence and Correctness.

Parameters Name	Silver Standard	T5-base	BART-large	LLaMA-3-8B
Adequacy	2.17	2.4	2.54	2.23
Fluency	2.44	2.6	2.54	2.45
Coherence	2.47	2.7	2.54	2.34
Correctness	2.47	2.8	2.6	2.82

TABLE 8.5: Human evaluation scores for summaries generated by three fine-tuned models (BART-large, T5-base, LLaMA-3-8B) and one additional model (PGM + Coverage + SciBERT). Scores are based on the number of times each model-generated summary was chosen as the most or least preferred.

Most / Least preferred	BART-large	T5-base	LLaMA-3-8B	PGM + Coverage +SciBERT
Most preferred	61↑	57	59	21
Least preferred	19	11	36	82↓

Fine-tuned **LLaMA-3-8B** model achieved the highest scores in automatic evaluations across all metrics and was also one of the preferred models in human evaluations. Its strong performance in automatic metrics is reflected in its high preference among annotators. Summary predicted by fine-tuned *BART-large* model is another highly preferred model according to human evaluations and ranks highly in automatic metrics evaluation shown in Table 8.3, suggesting a strong correlation between high automatic scores and human preference. fine-tuned *T5-base* model shows good performance in both automatic metrics and human preferences but is slightly less preferred than fine-tuned *LLaMA-3-8B* and *BART-large* models.

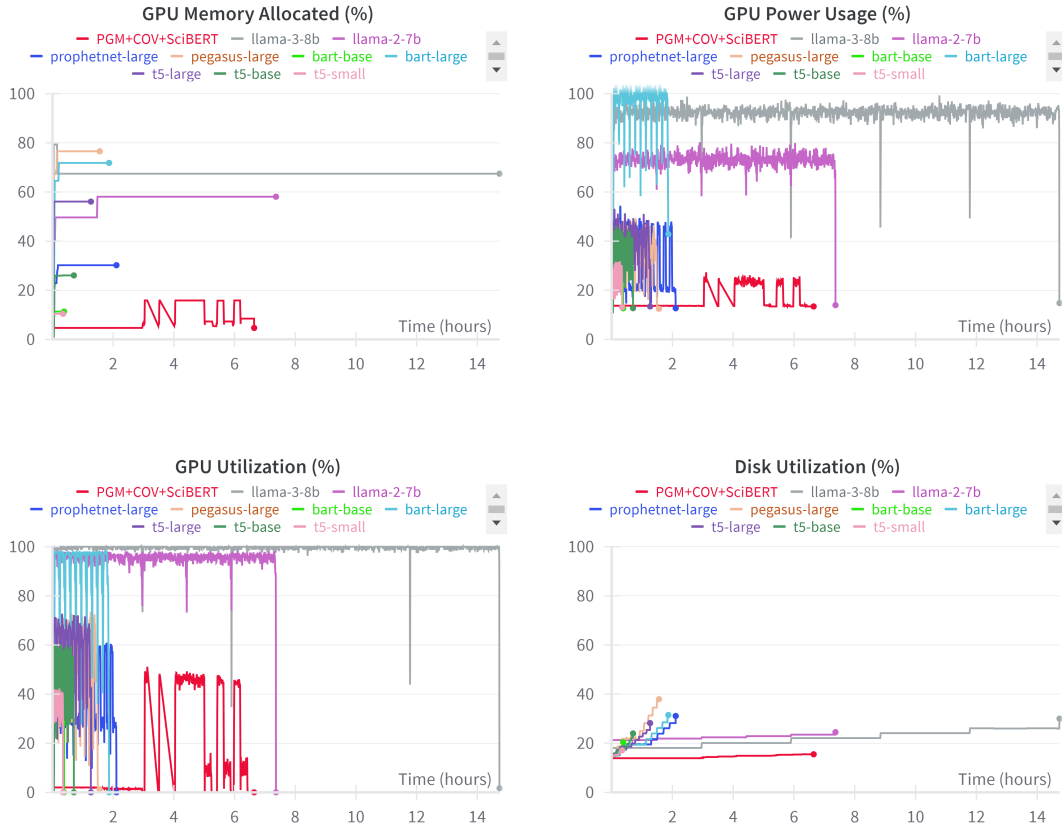


FIGURE 8.3: Comparison of compute resources used by summarization models.

8.9.2 Analysis of energy consumption

In this subsection, we discussed resource utilization during model execution using the WandB tool ⁶, focusing on memory usage, compute power consumption, and execution time. The fine-tuned LLaMA-3-8B model requires the highest GPU memory allocation (32 GB) and GPU power usage (399 Watts) and has the longest execution time compared to other models. Although, *LLaMA-3-8B* has 8 billion parameters, only a fraction is used, which affects its apparent parameter utilization. Figure 8.3 shows that *LLaMA-3-8B* initially allocates the most GPU memory and power and maintains continuous full GPU utilization throughout the execution. Additionally, while *LLaMA-2-7B* initially uses the most disk space, *Pegasus-large* eventually surpasses it.

⁶<https://wandb.ai/site>

8.10 Case Studies

Sample outputs from our models, generated using abstracts from the *SilverCSPicoSum* dataset, are shown in Figure 8.4, with repeating words highlighted in background. In Figure 8.4, the phrase “to preprocess face images” is repeatedly generated by the PGM + SciBERT model. While the (*PGM + Coverage + SciBERT*) model reduces repetition, it fails to include key information such as “Experimental results show CMU PIE, Extended Yale B, and CAS-PEAL-R1 databases,” which is crucial for demonstrating effectiveness. In contrast, the fine-tuned T5-small and T5-base models produce summaries that match the ground truth (Silver pico summary generated by *ChatGPT-3.5*).

8.11 Discussion

The fine-tuned *LLaMA-3-8B* model consistently outperforms others in both automatic evaluation metrics and is highly favored by human evaluators, showcasing a strong balance between technical accuracy and human preference. The fine-tuned *BART-large* model follows closely in automatic metrics but excels in human evaluations, reflecting its strong appeal despite slightly lower quantitative scores. The fine-tuned *T5-base* model, while delivering solid results, is slightly less preferred, likely due to task-specific differences in performance or output style.

Furthermore, the use of large language models (LLMs) offers a transformative solution to the challenge of limited high-quality datasets. LLMs can efficiently generate large-scale, high-quality datasets, reducing the need for manual annotation while maintaining output quality. This capability enables the creation of more diverse datasets for text summarization, enhancing the generalization and performance of deep learning models across various domains and tasks.

8.12 Conclusion

We introduced the *SilverCSPicoSum* dataset, utilizing large language model *ChatGPT-3.5* for pico summary generation, which significantly reduces the cost and time of preparing human-annotated datasets. Our study demonstrates that while *LLaMA-3-8B* achieved the highest automatic evaluation scores on the *SilverCSPicoSum* dataset, fine-tuned *BART-large* model was the most preferred model in human evaluations, with *LLaMA-3-8B* showing similar but slightly lower preference levels. Future research should explore integrating human insights with AI models’ efficiency to enhance summarization quality, focusing on hybrid approaches that leverage both human and AI strengths.

ChatGPT-3.5 generated Pico Summary: “This paper proposes two methods using Local Histogram Specification to preprocess face images under varying lighting conditions, effectively removing high and low frequency illumination and enhancing face features. Experimental results demonstrate their effectiveness.”
PGM + SciBERT: “The paper introduces local histogram specification to preprocess face images using local histogram specification to preprocess face images under varying lighting conditions. It outperforms other methods in both experimental results. Experimental results demonstrate its effectiveness.”
PGM + SciBERT + Coverage : “ This paper proposes a methods for preprocess face images using local lighting conditions , improving frequency illumination and efficiency in low frequency parts. Experimental results show cmu pie, extended yale b and caspeal-r1 databases.”
T5-small: “The paper proposes two methods based on Local Histogram Specification (LHS) to preprocess face images under varying lighting conditions, removing low and high frequency illumination and enhancing weak face features. Experimental results demonstrate effectiveness and efficiency.”
T5-base: “The paper proposes two methods using Local Histogram Specification to preprocess face images under varying lighting conditions, significantly removing low and high frequency parts of illumination and enhancing weak face features. Experimental results demonstrate their effectiveness.”
T5-large: “ In this paper, we propose two methods to preprocess face images under varying lighting conditions. The proposed methods are able to significantly remove both the low and high frequency parts of illumination on face images. Experimental results demonstrate the effectiveness and efficiency of our methods.”
BART-base: “This paper proposes two methods using Local Histogram Specification to preprocess face images under varying lighting conditions, effectively removing low and high frequency illumination and enhancing weak features. Experimental results demonstrate their effectiveness.”
BART-large: “This paper proposes two methods using Local Histogram Specification to preprocess face images under varying lighting conditions, effectively removing low and high frequency illumination and enhancing weak face features. Experimental results demonstrate their effectiveness.”
ProphetNet-large: “This paper proposes two methods using local histogram specification to preprocess face images, effectively removing low and high frequency illumination and enhancing face features. Experimental results demonstrate their effectiveness and efficiency.”
Pegasus-large: “This paper proposes two methods using Local Histogram Specification (LHS) to preprocess face images under varying lighting conditions, removing low and high frequency illumination and enhancing face features. Experimental results show effectiveness and efficiency.”
Fine-tuned LLaMA-2-7B: “This paper proposes two methods using Local Histogram Specification (LHS) to preprocess face images under varying lighting conditions, effectively removing low and high frequency illumination while enhancing facial features.”
Fine-tuned LLaMA-3-8B: “The paper proposes two methods using Local Histogram Specification (LHS) to preprocess face images under varying lighting conditions, effectively removing low and high frequency illumination and enhancing face features. ”

FIGURE 8.4: Comparison of short summaries generated from an abstract by different models. Abstract taken from <https://www.sciencedirect.com/science/article/pii/S026288561400047X>.

9

Conclusion and Future Scope

The exponential growth of research publications makes it very challenging for scholars to keep up with the latest developments, even within narrow sub-fields. To address this, research articles often include highlights that concisely summarize key contributions. These highlights not only help researchers quickly grasp the core findings without reading the entire paper but also enhance discoverability by improving search engine accessibility, which can lead to increased recognition and citations.

Effective titles also play a crucial role in document summarization by providing an initial overview and impacting the article’s visibility. In long-document summarization, one major challenge is reducing hallucinations—where models generate inaccurate or irrelevant information—making accuracy in summaries essential for preserving research integrity.

Creating large, high-quality datasets for training deep learning models is time-consuming and costly. To mitigate this, large language models (LLMs) can be leveraged to automatically generate silver-standard datasets. These datasets, while not manually curated, provide a scalable and cost-effective solution for training models in document summarization tasks.

Therefore, this thesis focuses on abstractive summarization systems to generate research highlights or key findings from scientific documents. It also emphasizes the importance of the *MixSub* dataset for generating research highlights based on papers from various subject domains. In this thesis, we address the problem of hallucination

generation during the summarization of long scientific documents and the generation of highlights. Additionally, we tackle the issue of generating compact and effective titles for research papers to assist novice authors. Furthermore, we analyzed the challenges associated with the manual creation of datasets for training data-hungry deep learning models. The thesis also discusses these novel systems in details.

In this chapter, we first discuss the summary of the entire thesis and then discuss some future scopes with respect to the proposed models.

9.1 Summary

This thesis begins with a comprehensive discussion of text summarization systems, exploring their applications and the challenges over it. It then focuses on abstractive summarization within the context of scientific documents, identifying key difficulties and the scope of the task. Following this, the thesis clearly defines the problem statement and research gap, and outlines the primary objectives and motivations driving the research.

The thesis continues with an in-depth review of key prior research on automatic text summarization systems and provides an overview of currently available datasets. It also highlights the significance of evaluation metrics used in this field. The discussion includes the necessity for effective methods and datasets, addresses the challenges related to human-annotated datasets, and explores appropriate automatic metrics for assessing abstractive summarization systems. We then present the significant contributions of this research as follows:

Firstly, we aim at automatically generate research highlights from scientific documents. We propose abstractive summarization systems, including the different word embeddings such as **Glove**, **ELMo**, **SciBERT** with pointer generator model with coverage mechanism. This approach examines how various types of embeddings influence the performance of summarization models. By addressing this, the chapter aims to contribute to the advancement of summarization techniques and enhance the generation of research highlights, thereby assisting researchers in managing the increasing volume of scientific literature. The significance of the proposed approach is highlighted by its lightweight nature and broad applicability. Experimental results demonstrate its practical usability in major academic settings. However, there is room for improvement by incorporating additional contextual information, such as syntax and semantics. This is an area we intend to explore in future research.

Secondly, We solve the problem of dataset for research highlights generation from various domain from scientific documents. We contributed a dataset *MixSub* contains

abstract and author written research highlights from research papers from various domain. MixSub consists of research papers along with author-written highlights from different subject domains. The goal of this dataset is to offer a wider and more diverse selection of research topics, thereby strengthening the effectiveness and adaptability of summarization models. By providing a comprehensive and varied dataset, MixSub aims to enhance the performance and relevance of summarization systems, benefiting the broader research community.

Thirdly, a novel approach is introduced to address this challenge by integrating named entity recognition (NER) with abstractive summarization techniques. This method treats **multi-word named entities** as single units during the highlights generation process, aiming to prevent common issues such as entity fragmentation and misrepresentation. In scientific research papers, NER becomes even more complex due to the varied patterns of named entities across different domains. Accurately handling **multi-word named entities** is a significant challenge, as these entities often consist of specialized and intricate terms that can be misrepresented if not processed correctly. The pointer-generator model, enhanced with NER and a coverage mechanism, has proven to be highly effective in producing high-quality highlights. This approach effectively tackles major issues in summarization, such as the accurate representation of multi-word entities.

Fourthly, we address key challenges in generating accurate research highlights and summaries, particularly focusing on hallucination in long-form text summarization. Using the Longformer Encoder-Decoder (LED) model, we generated summaries for the PubMed dataset and extracted research highlights from CSPubSum. Two techniques—data filtering and JAENS (Join sAlient ENtity and Summary generation)—were explored for their impact on factual consistency. We evaluated the summaries using entity-level precision-source and F1-target metrics, alongside traditional metrics like ROUGE, METEOR, MoverScore, and BERTScore. The fine-tuned LED model performed best on traditional metrics, while data filtering improved factual consistency. However, the JAENS approach under performed, requiring further investigation.

Fifthly, we focus on generating research paper titles from abstracts using deep neural models. Since crafting an effective title is challenging, automating this process can benefit authors. We fine-tuned several pre-trained transformer models, including *T5-base*, *BART-base*, and *PEGASUS-large*, on the CSPubSum dataset for this task. Additionally, we evaluated the open large language model *LLaMA-3-8B* with 8 billion parameters and compared its performance with and without fine-tuning. We also utilized *ChatGPT 3.5* in a zero-shot setting for title generation. Furthermore, we curated a new dataset, *LREC-COLING-2024*, of abstracts and titles to evaluate our fine-tuned

models.

Finally, to address the challenges of time-consuming and costly human-annotated dataset creation, we utilized pre-trained and large language models to develop a silver-standard dataset of abstracts with very short summaries. We introduced the *SilverCSPicoSum* dataset, generated from ScienceDirect papers using *GPT-3.5*, which reduces reliance on human annotation. Evaluations with the Precision-source metric indicated high precision and minimal hallucination in these summaries. Human assessments further confirmed their quality in terms of adequacy, fluency, coherence, and correctness. We fine-tuned several models, including *T5*, *BART*, *Pegasus*, and *ProphetNet*, on *SilverCSPicoSum*, and also fine-tuned *LLaMA-2* and *LLaMA-3* with LoRA. Additionally, we employed a pointer-generator network with a coverage mechanism, known for its high performance in summarization tasks. Model performance was assessed using ROUGE, METEOR, MoverScore, and BERTScore, along with human evaluations. Our study shows that while *LLaMA-3-8B* achieved the highest automatic evaluation scores on the *SilverCSPicoSum* dataset, fine-tuned *BART-large* model was most preferred in human evaluations, with *LLaMA-3-8B* showing slightly lower preference levels.

9.2 Future Scope

Finally, we discuss some of the future scope of the works presented in this thesis.

9.2.1 Multi-document Summarization Systems for Scientific Papers

While significant advancements have been achieved in both extractive and abstractive summarization of individual scientific documents, multi-document summarization in the academic realm remains underdeveloped. This is primarily due to the scarcity of labeled data and the inherent complexity of the task. Effective multi-document summarization necessitates not only a thorough understanding of each individual paper but also accurate modeling of the relationships between multiple papers. The Multi-XScience dataset [111] is one such resource available for this purpose. We aim to investigate multi-document summarization techniques utilizing this dataset.

9.2.2 Multi-lingual Research Highlight Generation for Scientific Papers

Our current research has been limited to scientific papers written in English. However, there is an increasing demand for research summaries to be available in multiple languages to accommodate researchers and readers who are more comfortable in their

native languages. Developing a mechanism for multi-lingual research highlights generation would greatly benefit non-native English speakers and novice researchers by making scientific papers more accessible and understandable.

9.2.3 Metrics for Abstractive Summarization Systems for Scientific Documents

Finally, we aim to enhance the evaluation of abstractive summarization systems. This involves developing automatic evaluation metrics to accurately identify all entities within scientific documents, ensuring that summaries are free from hallucinations. Additionally, these metrics should assess the abstractiveness of the generated research highlights, providing a comprehensive measure of their quality and relevance, and correlating with human judgments. Although some metrics such as GPTScore [59] to assess quality, they remain dependent on large language models, which may not always be reliable.

References

- [1] ACHIAM, J., ADLER, S., AGARWAL, S., AHMAD, L., AKKAYA, I., ALEMAN, F. L., ALMEIDA, D., ALTENSCHMIDT, J., ALTMAN, S., ANADKAT, S., ET AL. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023). [138](#)
- [2] AI@META. Llama 3 model card. [123](#), [145](#)
- [3] AKSENOV, D., MORENO-SCHNEIDER, J., BOURGONJE, P., SCHWARZENBERG, R., HENNIG, L., AND REHM, G. Abstractive text summarization based on language model conditioning and locality modeling. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (Marseille, France, May 2020), European Language Resources Association, pp. 6680–6689. [25](#)
- [4] AL-RADAIDEH, Q. A., AND BATAINEH, D. Q. A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms. *Cognitive Computation* 10 (2018), 651–669. [27](#)
- [5] ALAMI, N., MALLAHI, M. E., AMAKDOUF, H., AND QJIDAA, H. Hybrid method for text summarization based on statistical and semantic treatment. *Multimedia Tools and Applications* 80 (2021), 19567–19600. [27](#)
- [6] ALGULIEV, R. M., ALIGULIYEV, R. M., HAJIRAHIMOVA, M. S., AND MEHDIYEV, C. A. Mcmr: Maximum coverage and minimum redundant text summarization model. *Expert Systems with Applications* 38, 12 (2011), 14514–14522. [22](#)
- [7] ALGULIYEV, R. M., ALIGULIYEV, R. M., ISAZADE, N. R., ABDI, A., AND IDRIS, N. Cosum: Text summarization based on clustering and optimization. *Expert Systems* 36, 1 (2019), e12340. [22](#)
- [8] ALTMAMI, N. I., AND MENAI, M. E. B. Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences* 34, 4 (2022), 1011–1028. [28](#)
- [9] AMATO, F., MOSCATO, V., PICARIELLO, A., SPERLÍ, G., D’ACIERNO, A., AND PENTA, A. Semantic summarization of web news. *Encyclopedia with Semantic Computing and Robotic Intelligence* 1, 01 (2017), 1630006. [19](#)
- [10] AMMAR, W., GROENEVELD, D., BHAGAVATULA, C., BELTAGY, I., CRAWFORD, M., DOWNEY, D., DUNKELBERGER, J., ELGOHARY, A., FELDMAN, S., HA, V., KINNEY, R., KOHLMEIER, S., LO, K., MURRAY, T., OOI, H.-H., PETERS, M., POWER, J., SKJONSBORG, S., WANG, L. L., WILHELM, C., YUAN, Z., VAN

- ZUYLEN, M., AND ETZIONI, O. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)* (New Orleans - Louisiana, June 2018), Association for Computational Linguistics, pp. 84–91. 59
- [11] ANH, D. T., AND TRANG, N. T. T. Abstractive text summarization using pointer-generator networks with pre-trained word embedding. In *Proceedings of the 10th International Symposium on Information and Communication Technology* (2019), Association for Computing Machinery, pp. 473–478. 24, 140
- [12] ANTQUEIRA, L., OLIVEIRA JR, O. N., DA FONTOURA COSTA, L., AND NUNES, M. D. G. V. A complex network approach to text summarization. *Information Sciences* 179, 5 (2009), 584–599. 21
- [13] AONE, C., OKUROWSKI, M. E., GORLINSKY, J., AND LARSEN, B. A scalable summarization system using robust nlp. In *Intelligent Scalable Text Summarization* (1997). 19
- [14] ATRI, Y. K., GOYAL, V., AND CHAKRABORTY, T. Fusing multimodal signals on hyper-complex space for extreme abstractive text summarization (tl; dr) of scientific contents. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2023), pp. 3724–3736. 139
- [15] AZADANI, M. N., GHADIRI, N., AND DAVOODIJAM, E. Graph-based biomedical text summarization: An itemset mining and sentence clustering approach. *Journal of biomedical informatics* 84 (2018), 42–58. 5
- [16] BAHDANAU, D., CHO, K., AND BENGIO, Y. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations* (2015). 23, 45, 121, 139
- [17] BANERJEE, S., AND LAVIE, A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (2005), pp. 65–72. <https://aclanthology.org/W05-0909/>. 7, 8, 10, 30, 33, 44, 47, 93, 105, 109, 123, 128, 147
- [18] BANKO, M., MITTAL, V. O., AND WITBROCK, M. J. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics* (2000), pp. 318–325. 18
- [19] BARZILAY, R., AND ELHADAD, M. Using lexical chains for text summarization. In *Intelligent Scalable Text Summarization* (1997). 21
- [20] BAXENDALE, P. B. Machine-made index for technical literature—an experiment. *IBM Journal of research and development* 2, 4 (1958), 354–361. 17
- [21] BELTAGY, I., LO, K., AND COHAN, A. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods*

- in *Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 3615–3620. [12](#), [59](#), [128](#), [145](#)
- [22] BELTAGY, I., PETERS, M. E., AND COHAN, A. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020). [6](#), [105](#)
- [23] BHOLA, A., MULLAPUDI, J., KOLLIPARA, S., AND SANAKA, T. Text summarization based on ranking techniques. In *2022 5th International Conference on Contemporary Computing and Informatics (IC3I)* (2022), IEEE, pp. 1463–1467. [19](#)
- [24] BINWAHLAN, M. S., SALIM, N., AND SUANMALI, L. Fuzzy swarm diversity hybrid model for text summarization. *Information processing & management* *46*, 5 (2010), 571–588. [26](#)
- [25] BOJANOWSKI, P., GRAVE, E., JOULIN, A., AND MIKOLOV, T. Enriching word vectors with subword information. *Transactions of the association for computational linguistics* *5* (2017), 135–146. [24](#), [42](#)
- [26] BOLUKBASI, T., CHANG, K.-W., ZOU, J. Y., SALIGRAMA, V., AND KALAI, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems* *29* (2016). [41](#)
- [27] BORNMANN, L., HAUNSCHILD, R., AND MUTZ, R. Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases. *Humanities and Social Sciences Communications* *8*, 1 (2021), 1–15. <https://www.nature.com/articles/s41599-021-00903-w.pdf>. [1](#), [37](#), [88](#), [138](#)
- [28] BRANDOW, R., MITZE, K., AND RAU, L. F. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management* *31*, 5 (1995), 675–685. [18](#)
- [29] BRIN, S., AND PAGE, L. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* *30*, 1-7 (1998), 107–117. [21](#), [27](#)
- [30] BROWN, T., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J. D., DHARIWAL, P., NEELAKANTAN, A., SHYAM, P., SASTRY, G., ASKELL, A., ET AL. Language models are few-shot learners. *Advances in neural information processing systems* *33* (2020), 1877–1901. [124](#), [138](#)
- [31] BURGESS, C., SHAKED, T., RENSHAW, E., LAZIER, A., DEEDS, M., HAMILTON, N., AND HULLENDER, G. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning* (2005), pp. 89–96. [20](#)
- [32] CACHOLA, I., LO, K., COHAN, A., AND WELD, D. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 4766–4777. [31](#), [139](#)

- [33] CAGLIERO, L., AND LA QUATRA, M. Extracting highlights of scientific articles: A supervised summarization approach. *Expert Systems with Applications* 160 (2020), 113659. [29](#), [74](#), [75](#), [78](#)
- [34] CAO, Z., WEI, F., LI, W., AND LI, S. Faithful to the original: fact-aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (2018), AAAI’18/IAAI’18/EAAI’18, AAAI Press. [29](#), [103](#)
- [35] CARBONELL, J., AND GOLDSTEIN, J. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), pp. 335–336. [22](#)
- [36] CARICHON, F., NGOUMA, C., LIU, B., AND CAPOROSI, G. Objective and neutral summarization of customer reviews. *Expert Systems with Applications* (2024), 124449. [6](#)
- [37] CHENG, J., ZHANG, F., AND GUO, X. A syntax-augmented and headline-aware neural text summarization method. *IEEE Access* 8 (2020), 218360–218371. [20](#)
- [38] CHOPRA, S., AULI, M., AND RUSH, A. M. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (2016), pp. 93–98. [23](#)
- [39] CHOWDHURY, A., NARANG, S., DEVLIN, J., BOSMA, M., MISHRA, G., ROBERTS, A., BARHAM, P., CHUNG, H. W., SUTTON, C., GEHRMANN, S., ET AL. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113. <https://arxiv.org/abs/2204.02311>. [138](#)
- [40] CIACCIO, E. J. Use of artificial intelligence in scientific paper writing, 2023. [122](#)
- [41] COHAN, A., DERNONCOURT, F., KIM, D. S., BUI, T., KIM, S., CHANG, W., AND GOHARIAN, N. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 615–621. [28](#), [31](#), [105](#), [107](#), [139](#)
- [42] COHAN, A., AND GOHARIAN, N. Scientific document summarization via citation contextualization and scientific discourse. *International Journal on Digital Libraries* 19 (2018), 287–303. [28](#)
- [43] COLLINS, E., AUGENSTEIN, I., AND RIEDEL, S. A supervised approach to extractive summarisation of scientific papers. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)* (Vancouver,

- Canada, Aug. 2017), Association for Computational Linguistics, pp. 195–205. <https://aclanthology.org/K17-1021/>. 5, 29, 31, 46, 74, 75, 78, 81, 94, 105, 107, 125, 139, 142
- [44] CONROY, J. M., AND O’LEARY, D. P. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (2001), pp. 406–407. 19
- [45] CONTRACTOR, D., GUO, Y., AND KORHONEN, A. Using argumentative zones for extractive summarization of scientific articles. In *Proceedings of COLING 2012* (2012), pp. 663–678. 28
- [46] CORBETT, P., BATCHELOR, C., AND TEUFEL, S. Annotation of chemical named entities. In *Biological, translational, and clinical language processing* (Prague, Czech Republic, June 2007), K. B. Cohen, D. Demner-Fushman, C. Friedman, L. Hirschman, and J. Pestian, Eds., Association for Computational Linguistics, pp. 57–64. 90
- [47] DE VARGAS FEIJÓ, D., AND MOREIRA, V. P. Rulingbr: A summarization dataset for legal texts. In *Computational Processing of the Portuguese Language* (Cham, 2018), Springer International Publishing, pp. 255–264. 32
- [48] DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407. 18
- [49] DEJONG, G. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science* 3, 3 (1979), 251–273. 17
- [50] DEVLIN, J., CHANG, M.-W., LEE, K., AND TOUTANOVA, K. 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)* (2019), pp. 4171–4186. 25, 29, 44, 58, 65, 102, 121, 124
- [51] D’SOUZA, J., AND AUER, S. Computer science named entity recognition in the open research knowledge graph. In *From Born-Physical to Born-Virtual: Augmenting Intelligence in Digital Libraries* (Cham, 2022), Y.-H. Tseng, M. Katsurai, and H. N. Nguyen, Eds., Springer International Publishing, pp. 35–45. 90
- [52] EDMUNDSON, H. P. New methods in automatic extracting. *Journal of the ACM (JACM)* 16, 2 (1969), 264–285. <https://dl.acm.org/doi/10.1145/321510.321519>. 17, 19
- [53] EL-KASSAS, W. S., SALAMA, C. R., RAFAA, A. A., AND MOHAMED, H. K. Automatic text summarization: A comprehensive survey. *Expert systems with applications* 165 (2021), 113679. <https://www.sciencedirect.com/science/article/pii/S0957417420305030>. 2, 4, 18, 19, 22, 38, 89, 138

- [54] ERKAN, G., AND RADEV, D. R. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479. 21, 139
- [55] FABBRI, A., LI, I., SHE, T., LI, S., AND RADEV, D. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 1074–1084. 31
- [56] FABBRI, A. R., KRYŚCIŃSKI, W., MCCANN, B., XIONG, C., SOCHER, R., AND RADEV, D. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021), 391–409. 47
- [57] FABBRI, A. R., KRYŚCIŃSKI, W., MCCANN, B., XIONG, C., SOCHER, R., AND RADEV, D. SummEval: Re-evaluating Summarization Evaluation. *Transactions of the Association for Computational Linguistics* 9 (04 2021), 391–409. <https://aclanthology.org/2021.tacl-1.24.pdf>. 138
- [58] FRIEDRICH, A., VALEEVA, M., AND PALMER, A. LQVSumm: A corpus of linguistic quality violations in multi-document summarization. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)* (Reykjavik, Iceland, May 2014), N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds., European Language Resources Association (ELRA), pp. 1591–1599. 144
- [59] FU, J., NG, S.-K., JIANG, Z., AND LIU, P. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (Mexico City, Mexico, June 2024), K. Duh, H. Gomez, and S. Bethard, Eds., Association for Computational Linguistics, pp. 6556–6576. 157
- [60] GANESAN, K., ZHAI, C., AND HAN, J. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics* (2010), Association for Computational Linguistics, pp. 340–348. 23
- [61] GIDIOTIS, A., AND TSOU MAKAS, G. A divide-and-conquer approach to the summarization of long documents. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 3029–3040. 2
- [62] GOOGLE. Efficiency – data centers, 2021. <https://www.google.com/about/datacenters/efficiency>. 66
- [63] GOYAL, T., AND DURRETT, G. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 3592–3603. 29, 30, 103

- [64] GOYAL, T., LI, J. J., AND DURRETT, G. News summarization and evaluation in the era of gpt-3. *arXiv preprint arXiv:2209.12356* (2022). <https://arxiv.org/pdf/2209.12356>. 26, 140
- [65] GREENLEAF, G., MOWBRAY, A., KING, G., AND VAN DIJK, P. Public access to law via internet: The australian legal information institute. *JL & Inf. Sci.* 6 (1995), 49. 32
- [66] GRISHMAN, R., AND SUNDHEIM, B. M. Message understanding conference-6: A brief history. In *COLING 1996 volume 1: The 16th international conference on computational linguistics* (1996). 90
- [67] GRUSKY, M., NAAMAN, M., AND ARTZI, Y. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 708–719. 31
- [68] GUPTA, S., AND GUPTA, S. K. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications* 121 (2019), 49–65. 23
- [69] GUPTA, V., AND KAUR, N. A novel hybrid text summarization system for punjabi text. *Cognitive Computation* 8 (2016), 261–277. 27
- [70] GUPTA, V., AND LEHAL, G. S. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence* 2, 3 (2010), 258–268. 33
- [71] HARRIS, Z. S. Distributional structure. *Word* 10, 2-3 (1954), 146–162. 40
- [72] HASSEL, M. Exploitation of named entities in automatic text summarization for swedish. In *NODALIDA '03–14th Nordic Conference on Computational Linguistics, Reykjavik, Iceland, May 30–31 2003* (2003), p. 9. 89
- [73] HE, R., ZHAO, L., AND LIU, H. TWEETSUM: Event oriented social summarization dataset. In *Proceedings of the 28th International Conference on Computational Linguistics* (Barcelona, Spain (Online), Dec. 2020), D. Scott, N. Bel, and C. Zong, Eds., International Committee on Computational Linguistics, pp. 5731–5736. 32
- [74] HERMANN, K. M., KOCISKY, T., GREFFENSTETTE, E., ESPEHOLT, L., KAY, W., SULEYMAN, M., AND BLUNSOM, P. Teaching machines to read and comprehend. *Advances in neural information processing systems* 28 (2015). 30
- [75] HOU, L., HU, P., AND BEI, C. Abstractive document summarization via neural model with joint attention. In *Natural Language Processing and Chinese Computing: 6th CCF International Conference, NLPCC 2017, Dalian, China, November 8–12, 2017, Proceedings 6* (2018), Springer, pp. 329–338. 23

- [76] HU, B., CHEN, Q., AND ZHU, F. LCSTS: A large scale Chinese short text summarization dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), L. Màrquez, C. Callison-Burch, and J. Su, Eds., Association for Computational Linguistics, pp. 1967–1972. 32
- [77] IEA. Global energy & CO₂ status report 2019, 2019. <https://www.iea.org/reports/global-energy-co2-status-report-2019>. 66
- [78] JAIN, D., BORAH, M. D., AND BISWAS, A. Bayesian optimization based score fusion of linguistic approaches for improving legal document summarization. *Knowledge-Based Systems* 264 (2023), 110336. 5
- [79] JAMALI, H. R., AND NIKZAD, M. Article title type and its relation with the number of downloads and citations. *Scientometrics* 88, 2 (2011), 653–661. 121
- [80] JONES, K. S. Index term weighting. *Information storage and retrieval* 9, 11 (1973), 619–633. 41
- [81] KANG, D., AMMAR, W., DALVI, B., VAN ZUYLEN, M., KOHLMEIER, S., HOVY, E., AND SCHWARTZ, R. A dataset of peer reviews (PeerRead): Collection, insights and NLP applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 1647–1661. 31
- [82] KARYAKIN, A., AND SALEM, K. An analysis of memory power consumption in database systems. In *Proceedings of the 13th International Workshop on Data Management on New Hardware* (2017), pp. 1–9. 66
- [83] KAZANTSEVA, A., AND SZPAKOWICZ, S. Summarizing short stories. *Computational Linguistics* 36, 1 (2010), 71–109. 6, 28
- [84] KIM, B., KIM, H., AND KIM, G. Abstractive summarization of Reddit posts with multi-level memory networks. 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)* (Minneapolis, Minnesota, June 2019), J. Burstein, C. Doran, and T. Solorio, Eds., Association for Computational Linguistics, pp. 2519–2531. 32
- [85] KIM, M., MOIRANGTHEM, D. S., AND LEE, M. Towards abstraction from extraction: Multiple timescale gated recurrent unit for summarization. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (Berlin, Germany, 2016), Association for Computational Linguistics, pp. 70–77. 28
- [86] KINUGAWA, K., AND TSURUOKA, Y. A hierarchical neural extractive summarizer for academic papers. In *New Frontiers in Artificial Intelligence: JSAI-isAI Workshops, JURISIN, SKL, AI-Biz, LENLS, AAA, SCIDOCA, kNeXI, Tsukuba, Tokyo, November 13-15, 2017, Revised Selected Papers 9* (2018), Springer, pp. 339–354. 29

- [87] KLEINBERG, J. M. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46, 5 (1999), 604–632. 21
- [88] KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2* (San Francisco, CA, USA, 1995), IJCAI'95, Morgan Kaufmann Publishers Inc., p. 1137–1143. 64
- [89] KORNILOVA, A., AND EIDELMAN, V. Billsum: A corpus for automatic summarization of us legislation. *arXiv preprint arXiv:1910.00523* (2019). 32
- [90] KRYSCINSKI, W., MCCANN, B., XIONG, C., AND SOCHER, R. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Online, Nov. 2020), B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 9332–9346. 29, 30, 35, 103, 109, 128
- [91] KRYŚCIŃSKI, W., PAULUS, R., XIONG, C., AND SOCHER, R. Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Association for Computational Linguistics, pp. 1808–1817. 24
- [92] KUPIEC, J., PEDERSEN, J., AND CHEN, F. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1995), pp. 68–73. <https://dl.acm.org/doi/10.1145/215206.215333>. 4, 19, 28, 38, 138, 139
- [93] KUSNER, M., SUN, Y., KOLKIN, N., AND WEINBERGER, K. From word embeddings to document distances. In *International conference on machine learning* (2015), PMLR, pp. 957–966. 34, 128
- [94] LAN, Z. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019). 44
- [95] LANNELONGUE, L., GREALEY, J., AND INOUE, M. Green algorithms: quantifying the carbon footprint of computation. *Advanced Science* 8, 12 (2021), 2100707. 65, 66
- [96] LETCHFORD, A., MOAT, H. S., AND PREIS, T. The advantage of short paper titles. *Royal Society open science* 2, 8 (2015), 150266. 121, 126
- [97] LEV, G., SHMUELI-SCHEUER, M., HERZIG, J., JERBI, A., AND KONOPNICKI, D. TalkSumm: A dataset and scalable annotation method for scientific paper summarization based on conference talks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), A. Korhonen, D. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 2125–2131. 31, 139

- [98] LEWIS, M., LIU, Y., GOYAL, N., GHAZVININEJAD, M., MOHAMED, A., LEVY, O., STOYANOV, V., AND ZETTLEMOYER, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Online, July 2020), D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Association for Computational Linguistics, pp. 7871–7880. <https://aclanthology.org/2020.acl-main.703>. 25, 65, 123, 124, 138, 141, 145, 147
- [99] LI, P., LAM, W., BING, L., AND WANG, Z. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (Copenhagen, Denmark, Sept. 2017), M. Palmer, R. Hwa, and S. Riedel, Eds., Association for Computational Linguistics, pp. 2091–2100. 24
- [100] LIANG, Z., DU, J., AND LI, C. Abstractive social media text summarization using selective reinforced seq2seq attention model. *Neurocomputing* 410 (2020), 432–440. 2
- [101] LIN, C.-Y. Training a selection function for extraction. In *Proceedings of the eighth international conference on Information and knowledge management* (1999), pp. 55–62. 19
- [102] LIN, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out* (Barcelona, Spain, July 2004), Association for Computational Linguistics, pp. 74–81. "<https://aclanthology.org/W04-1013>". 7, 8, 9, 21, 22, 30, 33, 44, 47, 93, 103, 105, 109, 123, 128, 147
- [103] LIN, C.-Y., AND HOVY, E. Identifying topics by position. In *Fifth conference on applied natural language processing* (1997), pp. 283–290. 19
- [104] LIN, H., AND NG, V. Abstractive summarization: A survey of the state of the art. In *Proceedings of the AAAI conference on artificial intelligence* (2019), vol. 33, pp. 9815–9822. 30
- [105] LIN, Z. Why and how to embrace AI such as ChatGPT in your academic life. *Royal Society open science* 10, 8 (2023), 230658. 122
- [106] LIU, X., YIN, D., ZHENG, J., ZHANG, X., ZHANG, P., YANG, H., DONG, Y., AND TANG, J. OAG-BERT: Towards a unified backbone language model for academic knowledge services. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), pp. 3418–3428. 121
- [107] LIU, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019). 44
- [108] LIU, Y., SHI, K., HE, K. S., YE, L., FABBRI, A. R., LIU, P., RADEV, D., AND COHAN, A. On learning to summarize with large language models as references. *arXiv preprint arXiv:2305.14239* (2023). 26, 140

- [109] LLORET, E., ROMÁ-FERRI, M. T., AND PALOMAR, M. Compendium: A text summarization system for generating abstracts of research papers. *Data & Knowledge Engineering* 88 (2013), 164–175. 5, 26, 28, 121
- [110] LOVINGER, J., VALOVA, I., AND CLOUGH, C. Gist: general integrated summarization of text and reviews. *Soft Computing* 23 (2019), 1589–1601. 22
- [111] LU, Y., DONG, Y., AND CHARLIN, L. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proc. of EMNLP* (Online, Nov. 2020), ACL, pp. 8068–8074. 156
- [112] LUHN, H. P. The automatic creation of literature abstracts. *IBM Journal of Research and Development* 2, 2 (1958), 159–165. <https://ieeexplore.ieee.org/document/5392672>. 17, 38, 121, 139
- [113] MA, C., WU, Z., WANG, J., XU, S., WEI, Y., LIU, Z., ZENG, F., JIANG, X., GUO, L., CAI, X., ET AL. An iterative optimizing framework for radiology report summarization with chatgpt. *IEEE Transactions on Artificial Intelligence* (2024). 5
- [114] MANI, I., AND BLOEDORN, E. Multi-document summarization by graph search and matching. *arXiv preprint cmp-lg/9712004* (1997). 20
- [115] MARCU, D. Improving summarization through rhetorical parsing tuning. In *Sixth Workshop on Very Large Corpora* (1998). 21
- [116] MAREK, P., MÜLLER, Š., KONRÁD, J., LORENC, P., PICHL, J., AND ŠEDIVÝ, J. Text summarization of Czech news articles using named entities. *arXiv preprint arXiv:2104.10454* (2021). 89
- [117] MASHECHKIN, I. V., PETROVSKIY, M., POPOV, D., AND TSAREV, D. V. Automatic text summarization using latent semantic analysis. *Programming and Computer Software* 37 (2011), 299–305. 18
- [118] MCKEOWN, K., AND RADEV, D. R. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval* (1995), pp. 74–82. 19
- [119] MCKEOWN, K. R., BARZILAY, R., EVANS, D., HATZIVASSILOGLU, V., KLAUVANS, J. L., NENKOVA, A., SABLE, C., SCHIFFMAN, B., AND SIGELMAN, S. Tracking and summarizing news on a daily basis with columbia’s newsblaster. In *Proceedings of the Second International Conference on Human Language Technology Research* (San Francisco, CA, USA, 2002), HLT ’02, Morgan Kaufmann Publishers Inc., p. 280–285. 5
- [120] MEENA, Y. K., AND GOPALANI, D. Evolutionary algorithms for extractive automatic text summarization. *Procedia Computer Science* 48 (2015), 244–249. 22

- [121] MEENA, Y. K., JAIN, A., AND GOPALANI, D. Survey on graph and cluster based approaches in multi-document text summarization. In *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)* (2014), IEEE, pp. 1–5. 22
- [122] MIAO, Y., AND LI, C. Wikisummarizer-a wikipedia-based summarization system. In *TAC* (2010), Citeseer. 21
- [123] MIHALCEA, R., AND CEYLAN, H. Explorations in automatic book summarization. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* (2007), pp. 380–389. 6
- [124] MIHALCEA, R., AND TARAU, P. A language independent algorithm for single and multiple document summarization. In *Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts* (2005). 21
- [125] MIKOLOV, T., SUTSKEVER, I., CHEN, K., CORRADO, G. S., AND DEAN, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems 26* (2013). 24, 50
- [126] MISHRA, P., DIWAN, C., SRINIVASA, S., AND SRINIVASARAGHAVAN, G. Automatic title generation for text with pre-trained transformer language model. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)* (2021), IEEE, pp. 17–24. 121
- [127] MOHAMMAD, S., DORR, B., EGAN, M., HASSAN, A., MUTHUKRISHNAN, P., QAZVINIAN, V., RADEV, D., AND ZAJIC, D. Using citations to generate surveys of scientific paradigms. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics* (2009), pp. 584–592. 28
- [128] MRIDHA, M. F., LIMA, A. A., NUR, K., DAS, S. C., HASAN, M., AND KABIR, M. M. A survey of automatic text summarization: Progress, process and challenges. *IEEE Access 9* (2021), 156043–156070. 38
- [129] NALLAPATI, R., ZHOU, B., DOS SANTOS, C., GULCEHRE, C., AND XIANG, B. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning* (Berlin, Germany, 2016), Association for Computational Linguistics, pp. 280–290. <https://aclanthology.org/K16-1028>. 23, 29, 30, 31, 45, 50, 57, 59, 93, 121
- [130] NAN, F., NALLAPATI, R., WANG, Z., NOGUEIRA DOS SANTOS, C., ZHU, H., ZHANG, D., MCKEOWN, K., AND XIANG, B. Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (Online, Apr. 2021), P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds., Association for Computational Linguistics, pp. 2727–2733. <https://>

- [//aclanthology.org/2021.eacl-main.235](https://aclanthology.org/2021.eacl-main.235). 11, 13, 29, 35, 103, 105, 106, 109, 110, 111, 123, 128, 129, 141, 143
- [131] NARAYAN, S., COHEN, S. B., AND LAPATA, M. 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* (Brussels, Belgium, Oct.-Nov. 2018), E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds., Association for Computational Linguistics, pp. 1797–1807. 30, 31, 103
- [132] NASEEM, U., RAZZAK, I., KHAN, S. K., AND PRASAD, M. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing* 20, 5 (2021), 1–35. 41
- [133] NENKOVA, A. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3* (2005), AAAI Press, pp. 1436–1441. 20
- [134] NENKOVA, A., MCKEOWN, K., ET AL. Automatic summarization. *Foundations and Trends® in Information Retrieval* 5, 2–3 (2011), 103–233. 121
- [135] NETO, J. L., FREITAS, A. A., AND KAESTNER, C. A. Automatic text summarization using a machine learning approach. In *Advances in Artificial Intelligence: 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002 Porto de Galinhas/Recife, Brazil, November 11–14, 2002 Proceedings 16* (2002), Springer, pp. 205–215. 20
- [136] NIKOLA NIKOLOV, M. P., AND HAHNLOSER, R. Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Paris, France, may 2018), European Language Resources Association (ELRA). 28
- [137] OSBORNE, M. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 workshop on automatic summarization* (2002), pp. 1–8. 20
- [138] PAICE, C. D. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In *Proceedings of the 3rd Annual ACM Conference on Research and Development in Information Retrieval* (GBR, 1980), SIGIR '80, Butterworth & Co., p. 172–191. 28
- [139] PATTERSON, D., GONZALEZ, J., LE, Q., LIANG, C., MUNGUIA, L.-M., ROTHCHILD, D., SO, D., TEXIER, M., AND DEAN, J. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350* (2021). 66
- [140] PAULUS, R., XIONG, C., AND SOCHER, R. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304* (2017). 24

- [141] PENNINGTON, J., SOCHER, R., AND MANNING, C. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, Oct. 2014), Association for Computational Linguistics, pp. 1532–1543. 12, 44, 45, 50
- [142] PENNINGTON, J., SOCHER, R., AND MANNING, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (2014), pp. 1532–1543. 42, 50
- [143] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, June 2018), M. Walker, H. Ji, and A. Stent, Eds., Association for Computational Linguistics, pp. 2227–2237. 12, 42, 49, 50, 51, 65
- [144] PRADHAN, A., AND TODI, K. Understanding large language model based metrics for text summarization. In *Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems* (Bali, Indonesia, Nov. 2023), D. Deutsch, R. Dror, S. Eger, Y. Gao, C. Leiter, J. Opitz, and A. Rücklé, Eds., Association for Computational Linguistics, pp. 149–155. 26
- [145] PRIYA, V., AND UMAMAHESWARI, K. Enhanced continuous and discrete multi objective particle swarm optimization for text summarization. *Cluster Computing* 22 (2019), 229–240. 22
- [146] PU, X., GAO, M., AND WAN, X. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558* (2023). <https://arxiv.org/pdf/2309.09558>. 26, 138, 140
- [147] QASEMIZADEH, B., AND SCHUMANN, A.-K. The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)* (2016), pp. 1862–1868. 90
- [148] QI, W., YAN, Y., GONG, Y., LIU, D., DUAN, N., CHEN, J., ZHANG, R., AND ZHOU, M. ProphetNet: Predicting future n-gram for sequence-to-SequencePre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 2401–2410. <https://aclanthology.org/2020.findings-emnlp.217>. 26, 141, 145, 147
- [149] RADEV, D., ALLISON, T., BLAIR-GOLDENSOHN, S., BLITZER, J., ÇELEBI, A., DIMITROV, S., DRABEK, E., HAKIM, A., LAM, W., LIU, D., OTTERBACHER, J., QI, H., SAGGION, H., TEUFEL, S., TOPPER, M., WINKEL, A., AND ZHANG, Z. MEAD - a platform for multidocument multilingual text summarization. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)* (Lisbon, Portugal, May 2004), M. T. Lino, M. F. Xavier,

- F. Ferreira, R. Costa, and R. Silva, Eds., European Language Resources Association (ELRA). 139
- [150] RADEV, D. R., BLAIR-GOLDENSOHN, S., AND ZHANG, Z. Experiments in single and multidocument summarization using mead. In *First document understanding conference* (2001), pp. 1–7. <https://emunix.emich.edu/~wsverdlik/COSC562/ExperimentsinSingleandMulti.pdf>. 139
- [151] RADEV, D. R., JING, H., STYŚ, M., AND TAM, D. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938. 21
- [152] RADFORD, A., NARASIMHAN, K., SALIMANS, T., AND SUTSKEVER, I. Improving language understanding by generative pre-training. *OpenAI Blog* (2018). <https://openai.com/blog/language-unsupervised/>. 25, 43, 65, 138
- [153] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y., LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. <https://jmlr.org/papers/v21/20-074.html>. 25, 123, 124, 138, 141, 145, 147
- [154] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y., LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research (JMLR)* 21, 1 (2020), 5485–5551. 66
- [155] RAFFEL, C., SHAZEER, N., ROBERTS, A., LEE, K., NARANG, S., MATENA, M., ZHOU, Y., LI, W., AND LIU, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 1 (jan 2020). 121
- [156] RASCHKA, S. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808* (2018). 64
- [157] RAU, L. F., JACOBS, P. S., AND ZERNIK, U. Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing & Management* 25, 4 (1989), 419–428. 18
- [158] REHMAN, T., DAS, S., SANYAL, D. K., AND CHATTOPADHYAY, S. Abstractive text summarization using attentive gru based encoder-decoder. In *Applications of Artificial Intelligence and Machine Learning: Select Proceedings of ICAAAIML 2021*. Springer, 2022, pp. 687–695. 25
- [159] REHMAN, T., DAS, S., SANYAL, D. K., AND CHATTOPADHYAY, S. An analysis of abstractive text summarization using pre-trained models. In *Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2021* (2022), Springer, pp. 253–264. 26

- [160] REHMAN, T., DAS, S., SANYAL, D. K., AND CHATTOPADHYAY, S. An analysis of abstractive text summarization using pre-trained models. In *Proc. Int. Conf. Computational Intelligence, Data Science and Cloud Computing* (2022), Springer Nature Singapore, pp. 253–264. 29, 102
- [161] REHMAN, T., MANDAL, R., AGARWAL, A., AND SANYAL, D. K. Hallucination reduction in long input text summarization. In *Proceedings of the International Conference on Security, Surveillance and Artificial Intelligence: ICSSAI-2023* (2024), CRC Press, pp. 307–316. eBook ISBN 9781003428459. 101
- [162] REHMAN, T., SANYAL, D. K., AND CHATTOPADHYAY, S. Research highlight generation with elmo contextual embeddings. *Scalable Computing: Practice and Experience* 24, 2 (2023), 181–190. <https://www.scpe.org/index.php/scpe/article/view/2238>. 12, 49, 74, 75, 138, 140
- [163] REHMAN, T., SANYAL, D. K., CHATTOPADHYAY, S., BHOWMICK, P. K., AND DAS, P. P. Automatic generation of research highlights from scientific. In *2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE'21), collocated with JCDL'21* (2021). <https://ceur-ws.org/Vol-3004/paper10.pdf>. 12, 44, 74, 75, 138, 140
- [164] REHMAN, T., SANYAL, D. K., CHATTOPADHYAY, S., BHOWMICK, P. K., AND DAS, P. P. Generation of highlights from research papers using pointer-generator networks and scibert embeddings. *IEEE Access* 11 (2023), 91358–91374. <https://ieeexplore.ieee.org/document/10172215?denied=>. 12, 57, 68, 74, 75, 77, 78, 79, 138, 139, 142, 145
- [165] REHMAN, T., SANYAL, D. K., MAJUMDER, P., AND CHATTOPADHYAY, S. Named entity recognition based automatic generation of research highlights. In *Proceedings of the Third Workshop on Scholarly Document Processing (SDP 2022) collocated with COLING 2022* (Gyeongju, Republic of Korea, Oct. 2022), Association for Computational Linguistics, pp. 163–169. <https://aclanthology.org/2022.sdp-1.18>. 88, 138
- [166] ROSTAMI, F., MOHAMMADPOORASL, A., AND HAJIZADEH, M. The effect of characteristics of title on citation rates of articles. *Scientometrics* 98 (2014), 2007–2010. 121
- [167] RUDRA, K., GOYAL, P., GANGULY, N., IMRAN, M., AND MITRA, P. Summarizing situational tweets in crisis scenarios: An extractive-abstractive approach. *IEEE Transactions on Computational Social Systems* 6, 5 (2019), 981–993. 27
- [168] SAHBA, R., EBADI, N., JAMSHIDI, M., AND RAD, P. Automatic text summarization using customizable fuzzy features and attention on the context and vocabulary. In *2018 world automation congress (WAC)* (2018), IEEE, pp. 1–5. 27
- [169] SAHOO, D., BHOI, A., AND BALABANTARAY, R. C. Hybrid approach to abstractive summarization. *Procedia computer science* 132 (2018), 1228–1237. 27

- [170] SALTON, G., ALLAN, J., BUCKLEY, C., AND SINGHAL, A. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science* 264, 5164 (1994), 1421–1426. 18
- [171] SALTON, G., SINGHAL, A., MITRA, M., AND BUCKLEY, C. Automatic text structuring and summarization. *Information processing & management* 33, 2 (1997), 193–207. 20
- [172] SANDHAUS, E. The new york times annotated corpus. In *Philadelphia: Linguistic Data Consortium* (2008). 31
- [173] SANKARASUBRAMANIAM, Y., RAMANATHAN, K., AND GHOSH, S. Text summarization using wikipedia. *Information Processing & Management* 50, 3 (2014), 443–461. 21
- [174] SEE, A., LIU, P. J., AND MANNING, C. D. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017), pp. 1073–1083. <https://aclanthology.org/P17-1099.pdf>. 12, 23, 29, 44, 45, 49, 50, 51, 52, 57, 59, 62, 82, 92, 93, 95, 121, 139, 142, 145, 147
- [175] SHARMA, E., LI, C., AND WANG, L. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, July 2019), A. Korhonen, D. Traum, and L. Màrquez, Eds., Association for Computational Linguistics, pp. 2204–2213. 32
- [176] SHAZEER, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202* (2020). <https://arxiv.org/pdf/2002.05202>. 145
- [177] SILBER, H. G., AND MCCOY, K. F. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics* 28, 4 (2002), 487–496. 22
- [178] SONG, S., HUANG, H., AND RUAN, T. Abstractive text summarization using lstm-cnn based deep learning. *Multimedia Tools and Applications* 78, 1 (2019), 857–875. 24
- [179] SOUZA, C. M., MEIRELES, M. R. G., AND VIMIEIRO, R. A multi-view extractive text summarization approach for long scientific articles. In *2022 International Joint Conference on Neural Networks (IJCNN)* (2022), pp. 01–08. 28
- [180] SPARCK JONES, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation* 28, 1 (1972), 11–21. 18
- [181] STEINBERGER, J., JEZEK, K., ET AL. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM* 4, 93–100 (2004), 8. 18
- [182] STRUBELL, E., GANESH, A., AND MCCALLUM, A. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of*

- the Association for Computational Linguistics* (Florence, Italy, July 2019), Association for Computational Linguistics, pp. 3645–3650. 65, 66
- [183] SU, J., AHMED, M., LU, Y., PAN, S., BO, W., AND LIU, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* 568 (2024), 127063. <https://www.sciencedirect.com/science/article/pii/S0925231223011864>. 145
- [184] SUN, C., YANG, Z., WANG, L., ZHANG, Y., LIN, H., AND WANG, J. Biomedical named entity recognition using bert in the machine reading comprehension framework. *Journal of Biomedical Informatics* 118 (2021), 103799. 90
- [185] SUTSKEVER, I., VINYALS, O., AND LE, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Cambridge, MA, USA, 2014), NIPS’14, MIT Press, p. 3104–3112. 23, 121, 139
- [186] SVORE, K., VANDERWENDE, L., AND BURGESS, C. 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)* (2007), pp. 448–457. 20
- [187] TAN, J., WAN, X., AND XIAO, J. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *IJCAI* (2017), vol. 17, pp. 4109–4115. 121
- [188] TEUFEL, S., AND MOENS, M. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics* 28, 4 (2002), 409–445. 28
- [189] TOUVRON, H., LAVRIL, T., IZACARD, G., MARTINET, X., LACHAUX, M.-A., LACROIX, T., ROZIÈRE, B., GOYAL, N., HAMBRO, E., AZHAR, F., ET AL. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023). 123, 125
- [190] TOUVRON, H., MARTIN, L., STONE, K., ALBERT, P., ALMAHAIRI, A., BABAEI, Y., BASHLYKOV, N., BATRA, S., BHARGAVA, P., BHOSALE, S., ET AL. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023). 145
- [191] TU, Z., LU, Z., LIU, Y., LIU, X., AND LI, H. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 76–85. <https://aclanthology.org/P16-1008>. 45, 50, 51, 60, 93, 145
- [192] ULRICH, J., CARENINI, G., MURRAY, G., AND NG, R. Regression-based summarization of email conversations. In *Proceedings of the international AAAI conference on web and social media* (2009), vol. 3, pp. 334–337. 6

- [193] VALLEJO-HUANGA, D., MORILLO, P., AND FERRI, C. A dataset of attributes from papers of a machine learning conference. *Data in brief* 24 (2019), 103836. 121
- [194] VAN NOORDEN, R. Global scientific output doubles every nine years. *Nature news blog* (2014). 1, 7, 11, 37, 88, 138
- [195] VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, Ł., AND POLOSUKHIN, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017)* (2017), pp. 6000–6010. <https://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf>. 20, 25, 26, 43, 65, 102, 121, 124, 126, 145
- [196] VERMA, V. K., YADAV, A., AND JAIN, T. Key feature extraction and machine learning-based automatic text summarization. In *Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3* (2019), Springer, pp. 871–877. 20
- [197] VICTOR, S., ALBERT, W., COLIN, R., STEPHEN, B., LINTANG, S., ZAID, A., ANTOINE, C., ARNAUD, S., ARUN, R., MANAN, D., ET AL. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations* (2022). 138
- [198] VIJAY KUMAR, N., AND JANGA REDDY, M. Factual instance tweet summarization and opinion analysis of sport competition. In *Soft Computing and Signal Processing: Proceedings of ICSCSP 2018, Volume 2* (2019), Springer, pp. 153–162. 6
- [199] VISSER, W., AND WIELING, M. Sentence-based summarization of scientific documents: The design and implementation of an online available automatic summarizer. Report, last retrieved June 26, 2023, 2007. <http://www.martijnwieling.nl/files/wielingvisser05automaticsummarization.pdf>. 28
- [200] VÖLSKE, M., POTTHAST, M., SYED, S., AND STEIN, B. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization* (Copenhagen, Denmark, Sept. 2017), Association for Computational Linguistics, pp. 59–63. 32
- [201] WANG, M., WANG, X., AND XU, C. An approach to concept-obtained text summarization. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.* (2005), vol. 2, IEEE, pp. 1337–1340. 18
- [202] WANG, Q., ZHOU, Z., HUANG, L., WHITEHEAD, S., ZHANG, B., JI, H., AND KNIGHT, K. Paper abstract writing through editing mechanism. *arXiv preprint arXiv:1805.06064* (2018). 121

- [203] WANG, S., ZHAO, X., LI, B., GE, B., AND TANG, D. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE international congress on big data (BigData congress)* (2017), IEEE, pp. 305–312. [23](#)
- [204] WANG, S., ZHOU, W., AND JIANG, C. A survey of word embeddings based on deep learning. *Computing* 102, 3 (2020), 717–740. [40](#), [41](#)
- [205] WANG, Y., HOU, Y., CHE, W., AND LIU, T. From static to dynamic word representations: a survey. *International Journal of Machine Learning and Cybernetics* 11 (2020), 1611–1630. [40](#)
- [206] WEI, B., REN, X., ZHANG, Y., CAI, X., SU, Q., AND SUN, X. Regularizing output distribution of abstractive chinese social media text summarization for improved semantic consistency. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 18, 3 (2019), 1–15. [24](#)
- [207] WIBAWA, A. P., KURNIAWAN, F., ET AL. A survey of text summarization: Techniques, evaluation and challenges. *Natural Language Processing Journal* 7 (2024), 100070. [20](#)
- [208] WITBROCK, M. J., AND MITTAL, V. O. Ultra-summarization (poster abstract) a statistical approach to generating highly condensed non-extractive summaries. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (1999), pp. 315–316. [18](#)
- [209] XU, Y., AND MA, Y. Evolutionary neural architecture search combining multi-branch convnet and improved transformer. *Scientific Reports* 13, 1 (2023), 15791. [20](#)
- [210] YAN, E., AND ZHU, Y. Identifying entities from scientific publications: A comparison of vocabulary-and model-based methods. *Journal of Informetrics* 9, 3 (2015), 455–465. [89](#)
- [211] YANG, M., LI, C., SHEN, Y., WU, Q., ZHAO, Z., AND CHEN, X. Hierarchical human-like deep neural networks for abstractive text summarization. *IEEE Transactions on Neural Networks and Learning Systems* 32, 6 (2020), 2744–2757. [25](#)
- [212] YANG, M., WANG, X., LU, Y., LV, J., SHEN, Y., AND LI, C. Plausibility-promoting generative adversarial network for abstractive text summarization with multi-task constraint. *Information Sciences* 521 (2020), 46–61. [24](#)
- [213] YANG, Z. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* (2019). [44](#)
- [214] YASUNAGA, M., KASAI, J., ZHANG, R., FABBRI, A. R., LI, I., FRIEDMAN, D., AND RADEV, D. R. Scisummnet: a large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and*

- Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (2019), AAAI'19/IAAI'19/EAAI'19, AAAI Press. 31, 139
- [215] YE, S., CHUA, T.-S., AND LU, J. Summarizing definition from wikipedia. 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* (2009), pp. 199–207. 21
- [216] YEH, J.-Y., KE, H.-R., AND YANG, W.-P. ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications* 35, 3 (2008), 1451–1462. 21
- [217] YEH, J.-Y., KE, H.-R., YANG, W.-P., AND MENG, I.-H. Text summarization using a trainable summarizer and latent semantic analysis. *Information processing & management* 41, 1 (2005), 75–95. 22
- [218] ZHAI, M., WANG, X., AND ZHAO, X. The importance of online customer reviews characteristics on remanufactured product sales: Evidence from the mobile phone market on amazon. com. *Journal of Retailing and Consumer Services* 77 (2024), 103677. 6
- [219] ZHANG, B., AND SENNRICH, R. Root mean square layer normalization. In *Advances in Neural Information Processing Systems* (2019), H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc. 145
- [220] ZHANG, H., LIU, X., AND ZHANG, J. Summit: Iterative text summarization via chatGPT. In *The 2023 Conference on Empirical Methods in Natural Language Processing* (2023). 26, 140
- [221] ZHANG, J., LI, K., AND YAO, C. Event-based summarization for scientific literature in chinese. *Procedia Computer Science* 129 (2018), 88–92. 28
- [222] ZHANG, J., ZHAO, Y., SALEH, M., AND LIU, P. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the International Conference on Machine Learning (ICLR)* (2020), PMLR, pp. 11328–11339. 25, 121
- [223] ZHANG, J., ZHAO, Y., SALEH, M., AND LIU, P. J. Pegasus: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning* (2020), ICML'20, JMLR.org. <https://dl.acm.org/doi/abs/10.5555/3524938.3525989>. 123, 124, 125, 138, 141, 145, 147
- [224] ZHANG, T., KISHORE, V., WU, F., WEINBERGER, K. Q., AND ARTZI, Y. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, (ICLR 2020)* (2020), pp. 1–43. <https://openreview.net/pdf?id=SkeHuCVFDr>. 7, 8, 10, 30, 33, 44, 47, 93, 103, 105, 109, 123, 128, 147

- [225] ZHANG, Y., LI, D., WANG, Y., FANG, Y., AND XIAO, W. Abstract text summarization with a convolutional seq2seq model. *Applied Sciences* 9, 8 (2019), 1665. 24
- [226] ZHANG, Z., HAN, X., LIU, Z., JIANG, X., SUN, M., AND LIU, Q. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (2019), Association for Computational Linguistics, pp. 1441–1451. 25
- [227] ZHAO, W., PEYRARD, M., LIU, F., GAO, Y., MEYER, C. M., AND EGER, S. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Association for Computational Linguistics, pp. 563–578. <https://aclanthology.org/D19-1053>. 7, 8, 10, 30, 33, 34, 35, 44, 47, 93, 105, 109, 123, 128, 147
- [228] ZHAO, Z., COHEN, S. B., AND WEBBER, B. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (Online, Nov. 2020), T. Cohn, Y. He, and Y. Liu, Eds., Association for Computational Linguistics, pp. 2237–2249. 138
- [229] ZHOU, C., LI, Q., LI, C., YU, J., LIU, Y., WANG, G., ZHANG, K., JI, C., YAN, Q., HE, L., ET AL. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419* (2023). 49
- [230] ZHU, C., HINTHORN, W., XU, R., ZENG, Q., ZENG, M., HUANG, X., AND JIANG, M. Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Online, June 2021), K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, Eds., Association for Computational Linguistics, pp. 718–733. 29, 103

10

List of Acronyms

ATS	Automatic Text Summarization	1
NLP	Natural Language Processing	1
PSO	Particle Swarm Optimization	22
BERT	Bidirectional Encoder Representations from Transformers	25
T5	Text-To-Text Transfer Transformer	25
ProphetNet	Predicting Future N-gram for Sequence-to-Sequence Pre-training	25
BART	Bidirectional Auto-Regressive Transformers	25
PEGASUS	Pre-training with Extracted Gap-sentences for Abstractive	25
PLMs	Pre-trained Language Models	25
LLMs	Large Language Models	26
NLU	Natural Language Understanding	40
NLG	Natural Language Generation	40
NER	Named Entity Recognition	10
LSA	Latent Semantic Analysis	18
NB	Naive Bayes Classifier	19
LLaMA	Large Language Model Meta AI	26
RNNs	Recurrent Neural Networks	29
OOV	out-of-vocabulary	29
RQ	Research Question	16
BoW	Bag-of-Words	40
CBOW	Continuous Bag of Words	41

GloVe Global Vectors for Word Representation	12
ELMo Embeddings from Language Models	12

Tahida Lehman
25/09/2024

Samiran Chattopadhyay
25/09/2024

PROFESSOR
Deptt. of Information Technology
JADAVPUR UNIVERSITY
Block -LB, Plot-8, Sector-3
Salt Lake, Kolkata-700106, India

Debarshi Kumar Sanyal
25/09/2024

DR. DEBARSHI KUMAR SANYAL
Assistant Professor
School of Mathematical and Computational Sciences
Indian Association for the Cultivation of Science
Kolkata-700 032, India