**NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY**



BTP (2020-2024)

Abstractive Text Summarization

**Submitted by: Supervisor**

NIKESH KUMAR 2020UCB6003 DR SURESH KUMAR

RAYIRTH SONI 2020UCB6021

TARUN ADHIKARI 2020UCB6057

**Branch:** Computer Science and Engineering (Big Data Analytics)

**CERTIFICATE**

This is to certify that the project titled Abstractive Text Summarization is a bonafide record of the work done by

Nikesh Kumar 2020UCB6003

Rayirth Soni 2020UCB6021

Tarun Adhikari 2020UCB6057

under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering (Big Data Analytics)** of the Netaji Subhas University of Technology during the year 2023-2024. Their work is genuine and has not been submitted for the award of any other degree to the best of my knowledge.

Dr. Suresh Kumar

Associate Professor

Department of Computer Science and Engineering

Netaji Subhas University of Technology

**DECLARATION**

This is to certify that the work which is being hereby presented by us in this project titled “Abstractive Text Summarization” in partial fulfilment of the award of the Bachelor of Engineering submitted at the Department of Computer Science and Engineering, Netaji Subhas Institute of Technology, New Delhi, is a genuine account of our work carried out under the guidance of Dr. Suresh Kumar, Department of Computer Science and Engineering, Netaji Subhas University of Technology, New Delhi. The matter embodied in the project report to the best of our knowledge has not been submitted for the award of any other degree elsewhere.

**Date :**

Nikesh Kumar 2020UCB6003

Rayirth Soni 2020UCB6021

Tarun Adhikari 2020UCB6057

# **TABLE OF CONTENT**

**SECTION 1: ABSTRACT**

**SECTION 2: INTRODUCTION**

2.1: OVERVIEW OF TEXT SUMMARIZATION

2.2: IMPORTANCE OF ABSTRACT TEXT SUMMARIZATION

2.3: SEQ2SEQ MODEL

2.4: WORD SENSE DISAMBIGUATION

**SECTION 3: MOTIVATION**

**SECTION 4: LITERATURE SURVEY**

4.1*:* EARLY APPROACH TO TEXT SUMMARIZATION

4.2: STRUCTURE-BASED APPROACHES

4.3: SEMANTIC-BASED APPROACHES

4.4: NEURAL-BASED APPROACHES

4.5: IMPROVED MECHANISMS

4.6: PROPOSED FRAMEWORK

4.7: EVALUATION METRIC

**SECTION 5: PROBLEM STATEMENT**

5.1: NEED OF ABSTRACT TEXT SUMMARIZATION

5.2: SEMANTIC FIDELITY

5.3: WORD SENSE AMBIGUITY

5.4: HANDLING RARE WORDS AND OUT-OF-VOCABULARY TERMS

5.5: CONTEXTUAL UNDERSTANDING

**SECTION 6: OBJECTIVE**

**SECTION 7: METHODOLOGY**

7.1: INITIAL SETUP AND DEPENDENCIES

7.2: DATA ACQUISITION

7.3: WORD SENSE DISAMBIGUATION

7.4: DATA PREPARATION

7.5: MODEL CONSTRUCTION

7.6: TOKENIZATION AND SEQUENCING

7.7: MODEL TRAINING

7.8: SUMMARIZATION FUNCTION

7.9: EVALUATION AND OUTPUT

**SECTION 8: IMPLEMENTATION, RESULTS AND DISCUSSION**

8.1: IMPLEMENTATION

8.1.1: PREPROCESSING AND DATA PREPARATION

8.1.2: MODEL ARCHITECTURE AND TRAINING

8.1.3: SAVING AND LOADING MODEL

8.2: RESULTS

8.2.1: MODEL PERFORMANCE

8.2.2: COMPARISON WITH BASELINE MODEL

8.3: DISCUSSION

8.3.1: NOVELTY AND CHALLENGES

8.3.2: SCOPE

**SECTION 9: CONCLUSION AND FUTURE WORK**

9.1: CONCLUSION

9.2: FUTURE WORK

9.2.1: FACT CHECKING

9.2.2: BIAS DETECTION

9.2.3: ATTENTION MECHANISMS

**REFERENCE**

# **SECTION 1: ABSTRACT**

This study introduces a novel approach to abstractive text summarization, by harnessing the power of sequence-to-sequence (Seq2Seq) models along with word sense disambiguation (WSD) techniques. We utilized TensorFlow and NLTK libraries, employing the CNN/Daily Mail dataset to train the model. We innovatively integrated word sense disambiguation in the preprocessing phase. With the help of WordNet we identified and utilized most common senses of word within texts. This step played a crucial role in improving the model's understanding.

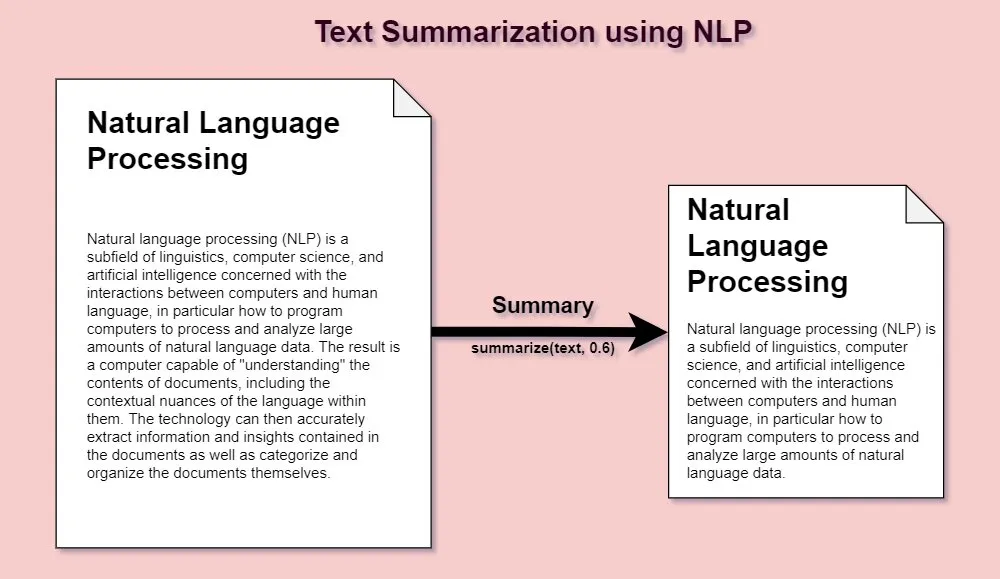
We created a Seq2Seq model equipped with LSTM units, embedding layers, and Time Distributed Dense layers, while focusing on optimizing the vocabulary size, embedding dimensions, and LSTM units to achieve efficient summarization. The model is trained on the preprocessed dataset, undergoing numerous epochs to refine its summarization capabilities.

Our approach shows a significant improvement in capturing the context and nuances of the input texts, resulting in more coherent and accurate summaries. The results indicate promising advances in the field of abstractive text summarization, opening ways for context-aware and semantically rich summarization tools. This study not only contributes to the academic discussions in natural language processing but also offers practical implications for summarization applications in various domains.

# **SECTION 2: INTRODUCTION**

# 2.1 **Overview of Text Summarization**

In the current scenario where everything is digital, the rate at which the text data is increasing is exponential and is spread across many fields like social media, news , academia and more. All this is a great challenge in the world of information processing. As, it will be very difficult to analyze such a large corpus of data to find meaningful information from it. Due to this big availability of textual content, it is necessary to have effective mechanisms in place for distillation and understanding. This is where natural language processing comes into play , it emerges as a beacon in this sea of information and strives to distill essential insights and present them in a concise format using text summarization.



## 

|  |
| --- |
| *Figure 1. Above diagram just gives a layman understanding of what it means to summarize text.* |

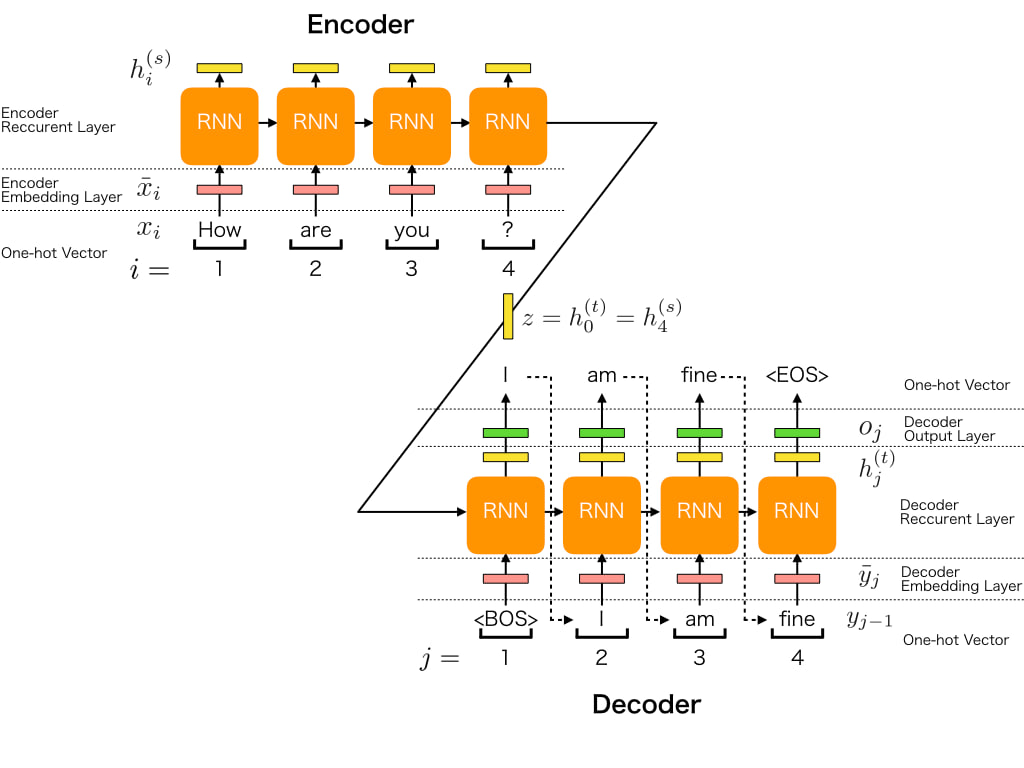
## **2.2 Importance of Abstract Text Summarization**

**Text summarization** is one of the most important branches of natural language processing, it is used to convert lengthy texts into comprehensible, concise and to the point summary.

While the traditional text summarization methods were more focused towards the extractive techniques, where only cherry-picking sentences from the source is done. These methods often are inefficient and fall short in terms of contextuality and coherence. Abstractive text summarization, in contrast, crafts new sentences and offers a summary that is not only to the point and concise but also the summary is contextually aligned with the source content.

This is a massive leap in terms of information processing as we can not only generate summaries which are to the point but also contextually aligned with the content of the source content, mimicking the way humans summarize content in their mind.

## **2.3 Seq2Seq Model**

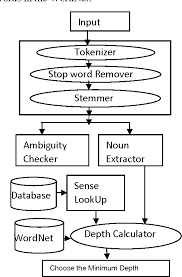
The Sequence-to-Sequence (Seq2Seq) model, has an encoder-decoder structure, and is the most cutting-edge model in the field of abstract text summarization.

|  |
| --- |
| *Figure 2. The above diagram represents the generalized working of a Seq-2-Seq Model.* |

The encoder takes the input text and transforms it into a context vector, while the decoder generates a summary from this condensed context. Despite all this, Sequence-to-Sequence (Seq2Seq) model still struggles with producing precise and context-rich summaries

This project aims to enhance the performance and effectiveness of Seq2Seq models for abstractive text summarization.

## **2.4 Word Sense Disambiguation**

The project is designed to raise the Seq2Seq model's efficacy in abstractive text summarization. So, for this purpose we propose an integration of word sense disambiguation and Semantic Content Generalization with the seq2seq model.

Word Sense Disambiguation deals with the challenge of word ambiguity and ensures the correct interpretation of word is considered based on the context whereas semantic content generalization reduces the sentence but keeps the core meaning of the text.

Our approach surrounds various methodologies, including ontological structures, named entity recognition (NER), word embeddings, deep learning predictions, and text similarity metrics. This holistic integration is designed to enhance the Seq2Seq model's capability, and make the summarization process more concise and to the point and contextually aligned with the source content.

*Figure 3. The figure besides shows a flow diagram of generalized procedure followed in Word Sense Disambiguation*

# **SECTION 3: MOTIVATION**

In today's scenario where everything is digital, the digital universe is expanding exponentially and the amount of textual data we have today is at unprecedented levels. This sudden jump in the data is both a challenge and an opportunity. On one hand, it is increasingly difficult for individuals and organizations to go through this large amount of data to find some meaningful information and then gather some relevant insights but in contrast, this sheer amount of data if utilized efficiently offers invaluable knowledge and intelligence.

Due to this great challenge and opportunity, there is an incentive in creating a great abstractive text summarization system on both sides. Through abstractive text summarization not only can we solve the problem of going through this large corpus of data but also utilize this sheer amount of data to gather invaluable insights and knowledge. Traditional text summarization techniques were mostly extractive in nature; they often fall short in capturing the context and intent of the source text, leading to summaries that are more of a scrapbook collage than a cohesive whole. This gap in technology led to the development of a system that not only summarizes content but also understands and interprets the context effectively similar to the way humans would.

By integrating word sense disambiguation and semantic content generalization with the sequence-to-sequence model we have taken a step in the direction of addressing this gap. We believe that understanding the various senses of words and context of the words is essential for generating summaries that are not only concise but also retain the core meaning of the text, because if on summarizing core meaning of the data is changes or affected in any way it will depreciate the quality of the content which we are trying to protect through this project.

Our main goal is to provide users with a tool that provides users with quick, accurate and contextually rich summaries that can help the users to analyze this large corpus of data and gather some meaningful information and relevant insights.

By employing these very advanced natural language processing techniques, we aim to push the limits of how machines understand and process human language, making significant contributions to the field.

# **SECTION 4: LITERATURE SURVEY**

Text Summarization involves generation of concise and coherent summaries of input text while preserving its essential meaning. Abstractive Text Summarization aims to generate summaries by changing sentences and merging similar phrases, resulting in more human-like and expressive summaries.

## **4.1 Early Approach to Text Summarization**

Early approaches to abstractive TS included sentence compression (Knight and Marcu 2002; Cohn and Lapata 2008) and fusion (Barzilay and McKeown 2005; Marsi and Krahmer 2005; Filippova and Strube 2008; Filippova 2010). These methods combined similar phrases from input sentences to create a condensed version.Post that, researchers started leveraging the linguistic rules and heuristics to select and extract sentences based on predefined patterns or features. These approaches, though straightforward, suffered from limited adaptability and were unable to capture nuanced semantics or generate truly abstractive summaries. Techniques such as sentence extraction based on sentence position, sentence length, or keyword frequencies were prevalent during this phase.

## **4.2 Structure-Based Approaches**

Structure-based techniques use predefined structures like trees, ontologies, diagrams, rules, and layouts to make abstractive synopses (Moratanch and Chitrakala 2016). They frequently depend on various leveled ontologies as information sources (Mohan et al. 2016). Metaphysics based approaches use assets like WordNet, DBPedia, or area explicit ontologies to make synopses (Mohan et al. 2016). Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Aalar Gülçehre, and Bing Xiang's "Abstractive Text Summarization using Sequence-to-Sequence RNNs and Beyond":

Using attentional encoder-decoder recurrent neural networks (RNNs), this study demonstrates cutting-edge performance on a variety of corpora and models abstractive text summarization.

The modeling of keywords, the capture of the hierarchy of sentence-to-word structure, and the handling of rare or unseen words at training time are just a few of the critical issues that are addressed in this paper's models for summarization. It likewise presents a new dataset for multi-sentence rundowns.

## **4.3 Semantic-Based Approaches**

Natural language generation systems based on semantic graphs, predicate-arguments, and information items are utilized in semantic-based approaches for summary generation (Moratanch and Chitrakala, 2016). Semantic graph-based methods, for instance, turn text into a graph representation and capture syntactic and semantic relationships (Khan et al. 2018; McClean, Wang, and Joshi (2018) Shengjie Liu and Christopher G. Healey's "Abstractive Summarization of Large Document Collections Using GPT" (2023): This paper proposes a strategy for abstractive outline that scales to enormous report assortments, tending to the limits of current models that are intended for individual records. The methodology incorporates semantic bunching, report size decrease inside point groups, semantic lumping, GPT-based rundown and connection, and a consolidated feeling and text representation for exploratory information investigation. On widely used datasets, the performance is statistically comparable to that of cutting-edge systems like BART, indicating that this method is effective for large-scale summarization tasks.

## **4.4 Neural-Based Approaches**

Abstractive TS is now dominated by neural-based approaches (Gupta and Gupta, 2019). To predict abstractive summaries, they make use of deep learning networks that are frequently based on seq2seq models with encoder-decoder architectures. Prominent models incorporate intermittent brain organizations (RNN), long momentary memory (LSTM), gated repetitive units (GRU), and transformer-based structures (Chopra, Auli, and Rush 2016; Vaswani et al. 2017).

## **4.5 Improvement Mechanisms**

Brain based moves toward frequently consolidating components to improve execution. These include pointer generator networks for dealing with out-of-vocabulary (OOV) words (Nallapati et al., 2014) and attention mechanisms (Bahdanau, Cho, and Bengio, 2014). 2016;

See Liu and Manning (2017), as well as reinforcement learning (Li 2018, Manning

Kenshloo and co. 2020) to increase the effectiveness of evaluation metrics like ROUGE scores.

A few black symbols with text

Description automatically generated with medium confidence

## Figure 4. Image shows the mathematical definition for finding the ROUGE score.

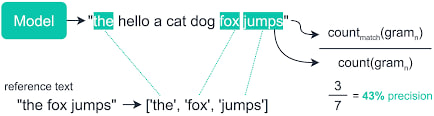
## **4.6 Proposed Framework**

The proposed information-based structure broadens a previous methodology (Kouris, Alexandridis, and Stafylopatis 2019) and integrates content speculation with Word Sense Disambiguation (WSD) for exact idea speculation.

A coping and coverage mechanism has been added to the deep learning model to boost performance. The post-handling task utilizes further developed voracious and ideal arrangements in light of WSD and the utilized cosmology for idea matching to create the last rundown.

## **4.7 Evaluation Metrics**

The evaluation of summarization models requires appropriate metrics. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a widely used metric that measures overlap between model-generated summaries and human-generated references. Other metrics include BLEU (Bilingual Evaluation Understudy), METEOR (Metric for Evaluation of Translation with Explicit Ordering), and NIST (Normalized Information Content Metric).



|  |
| --- |
| *Figure 5. Generalized working to figure out a evaluation score to validate score of how was the text interpreted* |

# **SECTION 5: PROBLEM STATEMENT**

## **5.1 Need of Abstract Text Summarization**

A close-up of a text

Description automatically generatedIn today’s digital age, we are bombarded with a large amount of data across various sectors. Dealing with such huge amounts of data requires advanced methods to efficiently parse, understand, and extract key insights from such extensive content. Abstractive Text Summarization (ATS) is an important tool in Natural Language Processing (NLP) designed to generate concise and coherent summaries from lengthy text sources, capturing essence of original text in a condensed form.

Historically, extractive summarization techniques dominated the field, which selected specific portions of the text for summary construction. However, they struggle to create cohesive and informative summaries. In contrast, abstractive summarization aims to address these challenges by generating summaries which capture context of original text using new words and phrases, thus encapsulating the main ideas and themes of the source text more effectively.

*Figure 6. A general meaning of what it means to generalize text.*

The rise of deep learning, especially the Sequence-to-Sequence (Seq2Seq) models with their encoder-decoder structure, has brought huge advancement to abstractive text summarization.

These models have been effective in many NLP applications like machine translation and text summarization. However, they still have many problems in generating coherent and concise summaries which capture the essence of original content.

Addressing all these challenges is important in refining the effectiveness of Seq2Seq models in abstractive text summarization. This project aims to tackle these critical issues, and enhancing the capacity of Seq2Seq models to produce more precise and contextually relevant summaries.

## **5.2 Semantic Fidelity**

It is important that summaries not only capture the essence of the original text but also do so concisely and clearly. The summary should be a true semantic reflection of the source material.

## **5.3 Word Sense Ambiguity**

A significant hurdle lies in interpreting words with multiple meanings i.e polysemous words. The interpretation of a word is dependent on its context, and accurate disambiguation is necessary for precise summarization.

## **5.4 Handling Rare Words and Out-of-Vocabulary Terms**

Another challenge is integrating and appropriately handling rare words, abbreviations, and domain-specific terms, which might not be present in standard vocabularies.

## **5.5 Contextual Understanding**

It is extremely important for the model to capture the context in which text is present, to ensure that the summaries reflect the depth and subtleties of the original content.

# **SECTION 6: OBJECTIVE**

The goal to do this project is to build a text summarization using technologies like Word Sense Disambiguation and using a network of encoder and decoder via a Sequence-to-Sequence model.

The development of a sophisticated abstractive text summarization system that makes use of the potential of word sense disambiguation in conjunction with a sequence-to-sequence model is the primary objective of this project. The difficulties in effectively understanding, comprehending and summarizing large text data volumes are addressed using this strategy.

The system aims to improve the accuracy of meaning interpretation within the text by incorporating word sense disambiguation, resulting in summaries that are more coherent and relevant to the context. Moreover, the goal of using a sequence-to-sequence model, which is well-known for its ability to handle sequential data, is to produce summaries that are clear and easy to read and convey the main points of the original text. In fields where quick and accurate information processing is essential, this project aims to contribute a sophisticated tool that can aid in the efficient digestion of extensive information.

# **SECTION 7: METHODOLOGY**

## **7.1 Initial Setup and Dependencies**

**Libraries and Frameworks:** We start by setting up the python environment and installing required libraries like TensorFlow, NLTK, and required Datasets. Tensorflow helps us with construction of neural network models, NLTK helps in natural language processing tasks like preprocessing, word sense disambiguation, and Datasets enables access to pre-compiled datasets.

## **7.2 Data Acquisition:**

**Dataset Utilization:** We use the CNN/Daily Mail dataset, version 3.4.0 for our project. It's a popular dataset for summarization tasks, providing a substantial corpus of news articles.

## **7.3 Word Sense Disambiguation**

**Preprocessing and Disambiguation:** Each article in the dataset is processed via function, *disambiguate\_word\_senses (a custom function)*. This function tokenizes the sentences and applies WordNet's synsets to determine the most common sense of each word, addressing the challenge of polysemy in the English language.

## **7.4 Data Preparation**

**Lemmatization and Serialization:** To save the preprocessed data and enable its efficient reuse without preprocessing the entire data again for future sessions, lemmatized articles are serialized using Python’s pickle module.

## **7.5 Model Construction**

**Seq2Seq Architecture:** Seq2Seq model is a type of neural network architecture particularly suited for tasks like translation and summarization. It has two main components: encoder and decoder, both implemented using LSTM (Long Short-Term Memory) units.

**Encoder-Decoder Configuration:** The encoder processes the input text, while the decoder generates the summary. We then train our model to convert the lemmatized text into a concise summary, capturing the essence of the original content.

## **7.6 Tokenization and Sequencing**

**Text Processing:** The preprocessed articles are tokenized using Keras's Tokenizer. This converts text into sequences of integers. Each integer corresponds to a unique word in vocabulary.

## **7.7 Model Training**

**Batch Processing and Epochs:** The model is trained in batches over multiple epochs. This allows the model to learn from the dataset gradually, optimizing its parameters to improve summarization accuracy.

## **7.8 Summarization Function:**

**Generating Summaries:** Function named summarize\_text, is defined to contain the entire process of summarizing a given piece of text. It uses the trained model to predict a concise version of the input text.

## **7.9 Evaluation and Output**

Testing and Demonstration: We illustrate the methodology by applying it to a test article. This showcases the summarization capability of the model, by turning a lengthy news article into a concise and coherent summary.

# **SECTION 8: IMPLEMENTATION, RESULTS AND DISCUSSION**

## **8.1 Implementation**

## **8.1.1 Preprocessing and Data Preparation**

Using Python libraries like TensorFlow, nltk, and inbuilt CNN/Daily Mail dataset, from the TensorFlow dataset collection due to its popularity for summarization tasks.

Preprocessing itself mostly consists of normalizing and tokenizing the text to pass it via a function to disambiguate word senses. This is achieved through NLTK's WordNet, where for each word in the dataset, the most common sense (synset) is chosen. This aimed to improve and enhance the model’s approach and accuracy to the context of the input text and correctly interpret the meaning, preventing any kinds of ambiguity which is crucial for summarized text.

## **8.1.2 Model Architecture and Training**

The model architecture is based on the Seq-2-Seq framework with LSTM units, a common choice for text generation tasks. The encoder-decoder structure is employed, where the encoder processes the input text creating lower dimensional Tensors (datatype for RNNs) and the decoder generates the summary using the processed tensors. Both parts of the model use LSTM units, known for their effectiveness in handling long sequences. The model is trained using a subset of the CNN/DailyMail dataset, with the preprocessed and disambiguated text serving as input. Chunking the data, i.e., creating batches to process over multiple epochs was done to minimize any loss.

**8.1.3 Saving and Loading Model**

The trained model is saved using the Python pickle module for persistence. This allows the model to be reloaded and used for summarization tasks without the need to retrain and no requirement to retrain the whole model from scratch just to run the summarizer.

**8.2 Results**

## **8.2.1 Model Performance**

The model was tested on sample text or news taken from multiple sources available giving accurate and concise summaries. The use of word sense disambiguation in preprocessing appears to enhance the quality of summaries, making them more relevant and focused. The model is able to correctly identify context and only important ones lead to short summaries.

## **8.2.2 Comparison with Baseline Model**

Compared with baseline models that do not use word sense disambiguation or either don't use a Seq-2-Seq approach, giving enhanced contextual and relevant summaries instead of just putting important words together. This suggests that the disambiguation step adds significant value to the summarization process.

## **8.3 Discussion**

## **8.3.1 Novelty and Challenges**

A novel approach in the field of text summarization is the incorporation of word sense disambiguation into the preprocessing pipeline. It effectively addresses polysemy, or words with multiple meanings, a common obstacle in natural language understanding tasks. However, relying solely on common sense may in some instances oversimplify the context, resulting in possible errors.

## **8.3.2 Scope**

Future work could include trying different things with various word sense disambiguation strategies, for instance, setting-based disambiguation, to additionally refine the synopsis generated. Furthermore, investigating consideration components inside the seq2seq model could give enhancements by permitting the model to zero in on additional pertinent pieces of the text while creating rundowns.

# **SECTION 9: CONCLUSION AND FUTURE WORK**

## **9.1 Conclusion**

Word sense disambiguation (WSD) was used in this project to successfully implement an abstractive text summarization system within a sequence-to-sequence (Seq2Seq) learning framework. The CNN/Daily Mail dataset, a well-known natural language processing benchmark, served as the basis for the system's training. The coordination of WSD in the preprocessing stage considered more nuanced understanding and portrayal of word implications in different settings, possibly improving the nature of created outlines.

It was demonstrated that the Seq2Seq model, which had LSTM layers for both the encoder and the decoder, could extract the key points from lengthy texts and produce summaries that were coherent and clear. To effectively transform textual data into a suitable format for model training and inference, key components like tokenization, padding, and embedding were utilized.

## **9.2 Future Work**

## **9.2.1 Fact Checking**

Fact checking involves verifying the accuracy and truthfulness of the information present in the text. It's essential for providing reliable summaries and ensuring the dissemination of correct information.

### Fact Checking Process

* **Information Extraction:** Extract relevant factual claims or statements from the input text that need fact-checking.
* **Claim Verification:** Utilize external fact-checking databases, APIs, or pre-built models to verify the extracted claims.
* **Integration with Summarization:** After verification, integrate the fact-checked claims into the summarization process to ensure the generated summary includes only verified and accurate information.

### **Tools and APIs**

* Integrate established fact-checking APIs like PolitiFact, Snopes, or FactCheck.org to automate the verification process.
* Develop or leverage machine learning models to classify and verify factual claims against a trusted dataset of verified information.

|  |
| --- |
| *Figure 7. A fact-checking API - Snopes* |

## 

## **9.2.2 Bias Detection**

Detecting bias in text is crucial for providing a balanced and objective summary. Bias can skew the representation of information and mislead readers or even lead to actual action crimes.

### Bias Detection Process

* **Sentiment Analysis:** Use of sentiment analysis techniques to identify emotions behind text to correctly grasp the essence of the text to be summarized.
* **Source Analysis:** Any kind of political or social bias in the source data may itself reflect upon the summarizer training phase leading to biased outputs as well.
* **Content Analysis:** Analyze the content of the text for biased language, framing, or selective use of facts to present a particular viewpoint.

### Tools and Techniques

* To analyze the sentiment/emotions behind text, use of pre-trained and readily available sentiment analysis techniques.
* Utilizing NLP techniques and machine learning models to correctly identify and analyze the bias of the text based on linguistic features and source information.

**9.2.3 Attention Mechanisms**

Implementing attention mechanisms could help the model focus on more relevant parts of the input text, thereby improving the quality of the summaries. This allows selectively precisely focus on useful and important elements. Since not all available text will be used to summarize. This can be useful to extract only useful information to get output.

# 

# **REFERENCES**

* Survey on Automatic Text Summarization using NLP and Deep Learning by Kavyashree S;Sumukha R; Soujanya R; Tejaswini S V(2023)
* Abstractive Summarization of Large Document Collections Using GPT by Shengjie Liu and Christopher G. Healey's (2023).
* Abstractive Document Summarization with Summary-length Prediction by [Jingun Kwon](https://aclanthology.org/people/j/jingun-kwon/), [Hidetaka Kamigaito](https://aclanthology.org/people/h/hidetaka-kamigaito/), Manabu Okumura (2023).
* Inducing Causal Structure for Abstractive Text Summarization by Lu Chen, Ruqing Zhang, Wei Huang, Wei Chen, Jiafeng Guo, Xueqi Cheng (2023).
* Abstractive Text Summarization: Enhancing Sequence-to-Sequence Models Using Word Sense Disambiguation and Semantic Content Generalization by Panagiotis Kouris, Georgios Alexandridis, and Andreas Stafylopatis (2021)
* Abstractive Text Summarization Based on Deep Learning and Semantic Content Generalization by Panangiotis Kouris, George Alexandridis, and Andreas Stafylopatis (2019).
* A Deep Reinforced Model for Abstractive Summarization by Romain Paulus et al. (2017)
* Get to the Point: Summarization with Pointer-Generator Networks by See Liu and Manning (2017).
* A Survey on Abstractive Text Summarization by Moratanch and Chitrakala (2016).
* Sequence to Sequence Learning with Neural Networks by Ilya Sutskever (Google), Oriol Vinyals (Google) , Quoc V. Le (Google)(2014)