Faster (Multi) Document Summarization Using PRIMERA & PEGASUS

Charavi Patil[1], Naman Shrimal[2], Priyal Saini[3], Shreya D Kashid[4], Urjasvi Kurakula[5]

[1,2,3,4,5]JAIN (Deemed to-be University), Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning)

Abstract

Text summarization is super important when it comes to dealing with loads of information that's available on the internet and in archives. Trying to summarize all that stuff manually would be impossible because there's just so much of it. That's why automatic summarization techniques have become a big deal. These techniques basically condense documents by picking out the most important ideas. At first, people were mainly focused on summarizing individual documents, but now the latest research is all about summarizing groups of documents. In our study, we've come up with this cool new way to combine the Primera and Pegasus models. This combo not only improves the quality of the summaries, but it also speeds up the whole process. And guess what? We can even summarize multiple documents at the same time thanks to some batch processing techniques that make everything run smoother.

Keywords: Text summarization, Automatic summarization, Extractive summarization, Abstractive summarization, Hybrid summarization, single-document summarization, Multi-document summarization, Natural Language Processing (NLP), Machine learning methods, Semantic representation, Sentence selection, Sentence scoring, Word frequency, Cross-domain summarization, Pretraining objectives.

1. Introduction

The quantity of records and facts at the Internet maintains to boom each day inside the form of internet pages, articles, educational papers, and news objects. In spite of the abundance, it's miles hard to discover records wanted correctly due to the fact maximum facts is irrelevant to a selected user's needs at a selected time. Therefore, the want for automated summarization and extraction of applicable facts is still a productive research region within natural language processing. Automatic summarization enables extract useful records even as discarding the inappropriate. It also can enhance the clarity of texts, and reduce the time that users spend in searching. Researchers were trying to perform appropriate automated text summarization for the reason that late Nineteen Fifties. The aim is to generate summaries of mul, combining the primary points in a readable and cohesive manner, while not having uncommon or repeated statistics [1].

Text summarization techniques typically extract crucial words, phrases or sentences from a report and use those phrases, phrases, or sentences to create a precis. Text summarization may be classified into unmarried file and multi-report summarization, relying on the number of enter documents. Single document text summarization handiest accepts one document as enter [2], whereas multi-record summarization accepts a couple of report, in which every report is related to the main subject matter.

Meaningful facts is extracted from each report and then gathered collectively and prepared to generate a precis [3] [4].

Extractive summarization chooses vital sentences from a report and combines them to create a precis without changing the original sentences. Abstractive summarization first converts the vital sentences extracted from a file into an understandable and coherent semantic form, after which generates the precis from this internal form, hence 1 potentially changing the authentic sentences. Hybrid textual content summarization combines each extractive and abstractive summarization.

Generally, the processing architecture of all computerized textual content summarization structures carries three steps. The first is preprocessing to typically identify phrases, sentences and different structural additives of the textual content. The 2d is processing, which converts the enter text to a precis through the use of a textual content summarization approach. The third is put up-processing, which fixes issues in the created draft precis [5]. Several recent surveys have been published on computerized text summarization, and most awareness on extractive summarization techniques [1] due to the fact abstractive summarization is hard and calls for comprehensive Natural Language Processing (NLP).

Most today's papers consciousness on part of automatic text summarization which includes specializing in one approach, or on one specific area in automated text summarization. We are the use of a hybrid gadget that combines extractive and abstractive summarization strategies to leverage their respective benefits. Therefore, the purpose of this survey is to present numerous techniques in text summarization to assist readers understand how an amazing summary can be generated by combining more than one approach or technique.

2. Motivation

2.1 Objective

The number one objective of this studies is to increase an innovative and efficient textual content summarization system the usage of a hybrid model that combines the power of the Primera and Pegasus fashions. This hybrid approach aims to generate brilliant summaries hastily, while harnessing the competencies of parallel processing. The research seeks to improve the sphere of automatic textual content summarization by way of presenting a unique and powerful answer for each single and multi-report summarization.

2.2 Motivation

Synergy of Pegasus and Primera Models: Our motivation to combine the Pegasus and Primera fashions is rooted of their unique strengths and abilities. Pegasus is renowned for its abstractive summarization talents, allowing for greater herbal and coherent summaries, while Primera excels in extractive summarization, imparting the benefit of preserving the supply text's wording and structure. The fusion of these models leverages the great of both worlds, aiming to acquire a stability among comprehensiveness and coherence inside the generated summaries. This planned desire is driven with the aid of our aspiration to create a hybrid model that harnesses the strengths of numerous summarization techniques to provide superior effects in numerous contexts.

Versatility in Summarization: One of the important thing motivations in the back of combining the Pegasus and Primera models in our hybrid method is to make certain versatility in summarization. This hybrid version is designed to efficiently summarize both multidocument and single-file content. By unifying the competencies of these fashions, we aim to offer a complete answer which could cope with the summarization needs of numerous customers and content sorts, thereby improving the accessibility and software of information across a extensive range of programs and domains.

Batch processing stands proud as a especially efficient technique inside the context of herbal language processing duties, and this is obvious within the furnished code. The usage of the padding token ensures that sequences are uniformly sized inside a batch, that's a important detail for green version training and inference. Without uniformity, the computational overhead concerned in managing sequences of different lengths may want to prevent overall performance.

The conventional strategies of summarization were in improvement because the past due Nineteen Fifties, striving to create concise, cohesive, and coherent summaries. While latest research generally recognition on extractive summarization because of its relative simplicity, this study adopts a holistic method by way of integrating each extractive and abstractive techniques. By leveraging the strengths of each approach, this have a look at targets to pave the way for more effective and complete text summarization.

Moreover, the implementation of a international interest mask is a recreation-changer. This mask empowers the model to take into account the whole record all through summarization, making it specifically valuable whilst coping with lengthy files segmented into sections. By specializing in precise tokens like the record separators and the start token, the model creates summaries that encapsulate the complete context and structure of the input.

In essence, batch processing no longer simplest optimizes GPU utilization and parallel processing however also benefits from the inherent parallelism of contemporary hardware. This twin advantage makes it a powerful device for appreciably reducing processing time. By correctly managing more than one documents at once, batch processing underscores its efficiency and speed in comparison to the time-consuming serial processing of files.

3. Literature Review

The review is organized into three sections: a brief introduction to text summarization, text summarization approaches, and the Hybrid approach discussed in this paper.

3.1 Text Summarization Approaches

Conceptually, there are three approaches for text summarization, which are extractive, abstractive, and hybrid summarization. Within each approach, there are many methods and techniques. Every approach has some advantages and disadvantages. A brief overview of the approaches along with some specific methods are shown in Figure 3.1

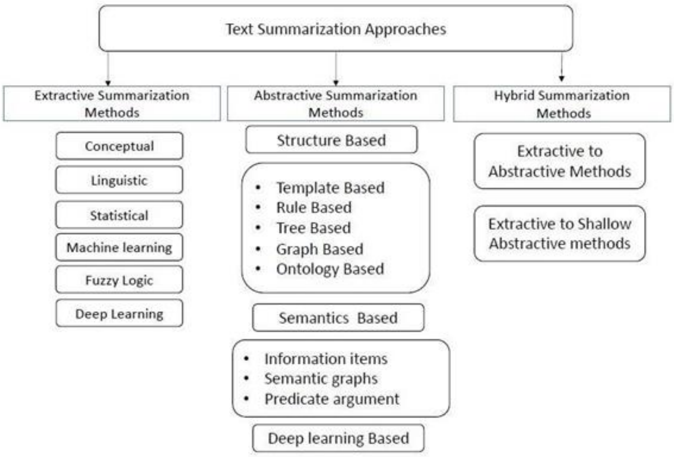


Figure 3.1: Text Summarization Approaches along with their Methods.

3.1.1 Extractive Summarization

The architecture for extractive summarization includes three steps: Pre-processing, Processing, and Post-processing, as shown in Figure 3.2. Pre-processing performs tasks such as tokenization and extraction of sentences and paragraphs. The processing step creates appropriate representation of the input text using techniques such as N-grams and graphs, or performs neural network based feature extraction and encoding [2] followed by scoring each sentence depending on input text representation [7]. After that, the approach chooses highly ranked sentences and links them together as a summary [7] [8]. Post-processing involves steps such as changing pronouns with their antecedents, and rearranging the extracted sentences [9].

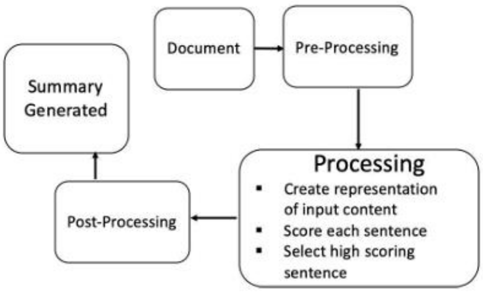


Figure 3.2: Extractive Text Summarization Architecture, Adopted from [5]

Advantages and Disadvantages for Extractive Summarization

Since extractive summarization depends on without delay producing the summary from the text without changing the content material sentences in any way, it is quicker and less complicated [10]. The disadvantage of this approach is that it is not the same as how human beings write the precis. The technique generally results in the reduction of semantic best and brotherly love because of wrong connections among sentences within the generated precis, making the go with the flow stilted and unnatural. The generated summary may not be accurate enough, and no longer cowl all critical content material sentences in the input report. However, if the output precis is lengthy enough, the difficulty of lacking large sentences might not arise. But it can include unnecessary components that might not be wished within the summary, making it longer than essential [9].

3.1.2 Extractive Summarization Methods

There are various extractive summarization methods for selecting and scoring sentences. These include Conceptual, Linguistic, Statistical, Machine Learning methods, Fuzzy logic, and Deep learning as presented in Figure 3.1.

Concept Methods

Such a method produces a summary of the concepts present in a document that can be found in external information repositories like WordNet [3] and Wikipedia. Depending on the concepts extracted, the important sentences are identified based on connection to external information bases instead of words. From the external information base's scores, a graph model or vector is built to produce the connection between the sentences and the concepts. The concept methods of summarization can cover a very large number of concepts because WordNet and Wikipedia are large repositories. However, such a method depends on high quality similarity measurements to decrease redundancies in calculating concept-sentence correlations[10].

Linguistic Methods

A linguistic method focuses on the relationships between words and concepts to get to the meaning to generate the summary. Abstractive summarization includes some level of semantic processing, so that, it can be thought also of as a linguistic method. Linguistic methods are useful because they try to understand the meaning of every sentence in a document. However, this method is time- consuming requiring high effort. A linguistic method also needs a large amount of memory for saving additional linguistic repositories such as WordNet. It needs powerful processors for complicated linguistic processing[14].

Statistical Methods

Such methods use statistical features of the document to identify the important pieces of the text. In a statistical method, a sentence is selected based on features like word frequency, position of the sentence, indicator phrases, title, location, and other features regardless of the meaning of the sentence. The method calculates the scores of the selected sentences and chooses a few highest scoring sentences to create the summary.

Baxendale [5] focused on the position of sentences in his summarization research. He found that the best locations for the most important parts of the paragraph are the first and last sentences. He examined 200 paragraphs, and concluded that the topic sentences are included the first sentence of the paragraph in around 85last sentence of the paragraph.

Statistical methods do not take into account the meaning of sentences, and as a result, they may produce low-quality summaries. Statistical methods require low memory and processor capacity.

Machine Learning Methods

The idea behind machine learning is to use a training set of data to train the summarization system, which is modeled as a classification problem. Sentences are classified into two groups: summary sentences and non-summary sentences. The probability of choosing a sentence for a summary is estimated according to the training documents and corresponding extractive summaries [18]. The steps for ranking sentences in Machine Learning methods are extracting features from a document, and feeding those features to a machine learning algorithm that gives an output score as a value. Some of the common machine learning methods used for text summarization are linear regression, naive Bayes, support vector machine, artificial neural networks, and fuzzy logic.

A large training data set is necessary to improve the choices of sentences for the summary [2]. A simple regression model may be able to produce better output when compared with the other classifiers [5]. Every sentence in the basic text must be labelled as a summary or 8 non-summary, demanding extensive manual work to generate extractive summaries for training [12].

Fuzzy Logic Based Methods

Such text summarization methods use a multiple-valued system known as fuzzy logic. Fuzzy logic produces an efficient way to provide feature values for sentences that are between the two logical values “one” and “zero”, because these two values often do not represent the “real world” [2]. For ranking sentences, the first step is to choose a group of features for every sentence. The second step is to apply the fuzzy logic concept to get a score for every sentence based on the importance of the sentence. This means every sentence has a score value from 0 to 1, depending on the features [5].

Fuzzy logic represents uncertainties in selecting a sentence as a ‘fuzzy’concept [1]. However, one negative factor is redundancy in the selected sentences for the summary, impacting the quality of the generated summary. Therefore, a redundancy removal technique is required to enhance the quality of the generated summary [1].

Deep Learning Methods

Kobayashi et al. [2] propose a device for text summarization using report level similarity relying on embeddings. They assume that an embedding of a phrase represents its that means, a sentence considered as a bag-of-phrases, and a report as a bag-of-sentences. They formalize their project because the trouble of maximizing a submodular function that is recognized with the aid of a negative summation of closest neighbors' distance on embedding distributions. They located that the report level similarity is greater complex in which means compared with sentence-degree similarity. In Chen et al. [3], they endorse computerized textual content summarization that used a reinforcement learning algorithm and Recurrent Neural Network (RNN) version with a unmarried document. By the use of a sentence degree selective encoding approach, they pick out the significant functions, generating the summary sentences.

In deep gaining knowledge of strategies, the network can be skilled depending at the reader's style, and the capabilities may be modified relying on the user's requirement. However, it's miles hard to perceive how the network generates a choice. Recent studies suggests that using a mixture of diverse methods allows produce a better summary through taking the advantage of the strengths of the person methods [1]. For example, Moratanch and Chitrakala used a mixture of each graphs and idea based strategies to generate summaries. Mao et al. [2] integrate 3 exclusive strategies of supervised learning with unsupervised learning to create a summary for a single record. Combining specific functions collectively can also help produce better results for the duration of the calculation of the weights of sentences [1].

3.1.3 Abstractive Summarization

Abstractive text summarization creates a precis of a file by means of extracting and knowledge the standards present inside the text for the duration of processing [7]. It paraphrases the text, but does no longer without delay reproduction from the content material of the original textual content [9]; instead it creates new sentences that better replicate the human way of building summaries. As a result, the enter content material needs more analysis for abstractive summarization [3].

The processing architecture for abstractive summarization. It is composed of Pre- processing, Processing that includes two sub-steps, and Post-processing. For instance, Moratanch and Chitrakala create an inner semantic representation and then use diverse techniques to create summaries [3].

Advantages and Disadvantages of Abstractive Summarization:

Some of the blessings of abstractive summarization are that the generated precis is created to be one of a kind from the original text by using the usage of extra resilient expressions based on paraphrasing [1]. So, the generated precis is probably to be towards a human summary [2]. Compared to extractive summarization, abstractive summarization can lower the quantity of generated text and bring a summary that removes any redundancy, acquiring a concise and expressive summary [3].

Some of the risks of abstractive summarization are that it is tough to perform superb abstractive summarization [1]. It is hard to create an awesome abstractive summary as it needs to use natural language era generation, which still desires loads of progress [3]. Current abstractive summarization approaches appear to create repetitions in word preference. In addition, appropriate abstractive summarization must be able to explain why it creates new sentences within the precis, that's hard to do. The approach is likewise not able to address out-of-vocabulary phrases properly [1]. Furthermore, the approach's capacity is constrained via what underlying semantic illustration it makes use of, due to the fact a gadget can't generate a summary if its representation scheme can not seize necessary nuances and info [9].

3.1.4 Abstractive Summarization Methods

Abstractive summarization methods can be classified into three categories, which are structure-based, semantics-based, and deep learning-based methods [3]. A structure-based approach uses pre-defined structures such as trees, graphs, templates, rules, and ontologies. Therefore, it recognizes in the input document, the most important information, and then using the previously mentioned structures, it generates the abstractive summary. The semantics based construction of the input document generates a semantic representation by using information items, semantic graphs, and predicate- argument structures. Then, using approaches in natural language generation, it generates the abstractive summary [5].

1. **Structure-Based Methods:**

Templates-Based Methods

Human summaries tend to use certain characteristic sentence structures in some domains. These can be identified as templates. To perform abstractive summarization, the information in the input document is used to fill slots in appropriate pre-defined templates based on the input document's style [7]. Text snippets can be extracted using rules and linguistic cues, to fill template slots [10].

Rule-based Methods

To find the important concepts in the input document and use them in the generated summary, one needs to define rules and categories. To use these methods, one needs to classify the input document based on the concepts and terms present in it, create relevant questions depending on the domain of the input document, answer the questions by detecting the concepts and terms in the document, and feed the answers into patterns to generate the summary [1].

Tree-based Methods:

To perform abstractive summarization in tree-based methods, one needs to cluster similar sentences in the input that have related information, and then work with these sentence clusters for the summary [4]. Similar sentences are formulated into trees, parsers are applied to build the dependency trees, a popular tree based representation. Then, a process such as pruning linearization is used to produce trees in order to generate summary sentences from some of the sentence clusters [10].

Graph-Based Methods

The authors in [5] used a graph model which contains nodes, with each node expressing a word and positional information, that is connected to other nodes. The structure of sentences is represented by directed edges. The steps for the graph method contain constructing a textual graph representing the source document and generating abstractive summary. Such a method explores and scores many sub-paths in the graph in order to create the abstractive summary [7].

2. **Semantics-Based Methods**

These methods process the input text to obtain semantic representations such as information items, semantic graphs, and predicate-argument structures. The representation is processed to provide the abstractive summarization by performing word choices, and stringing the words together using verb and noun phrases [2]. The authors in [7] perform multi-document abstractive summarization by extracting predicate-argument structures from the input text by performing semantic role labeling. By using a semantic similarity measurement, they cluster the semantically similar predicate-argument structures in the text, and then score the predicate-argument structures using feature weighting. Finally, they use language generation approaches to create sentences from predicate-argument structures.

4.Research Gap

4.1 Cross-Domain Summarization:

Many existing summarization models are domain-specific. Developing techniques for cross-domain summarization is a pressing research gap that aims to create summarization models capable of generating coherent summaries across various topics and fields. – By addressing these research gaps, your research can contribute significantly to the field of automatic text summarization and provide practical solutions for more efficient and comprehensive information extraction.

4.2 Transition from Single-Document to Mult Document Summarization:

In the realm of automatic text summarization, a prominent challenge is the generation of concise and coherent summaries from single documents. Traditional approaches often yield summaries sequentially, one document at a time, which can be highly inefficient and impractical in the face of the vast and ever-growing volume of digital content. This research gap emphasizes the necessity to advance from single-document summarization to more versatile multi-document summarization techniques.

While single-document summarization serves a valuable purpose, it falls short in scenarios where a comprehensive understanding of a topic requires information from multiple sources. In an interconnected digital landscape, users often need to synthesize insights and knowledge from various documents related to a specific topic or event. Transitioning to multi-document summarization addresses this gap by enabling the extraction of salient information from a collection of documents and presenting it in a cohesive manner. The research will explore innovative methods and models to facilitate this transition, ensuring that users can obtain comprehensive summaries when dealing with multiple, interconnected documents.

4.3 Accelerating Multi-Document Summarization:

While the shift to multi-document summarization addresses the challenge of information synthesis, it introduces another crucial concern—processing efficiency. Generating summaries from multiple documents concurrently can be a computationally demanding task, potentially leading to extensive processing times.

Recognizing this issue, a significant research gap lies in the development and implementation of batch processing techniques for multi-document summarization. Batch processing, a well-established concept in various computing domains, has the potential to significantly enhance the efficiency of summarization systems. However, its integration into the context of summarization is an unexplored area.

This research will focus on the design and implementation of batch processing strategies tailored to the demands of multi-document summarization. By harnessing the parallel processing capabilities of modern computing architectures, the aim is to drastically reduce processing times without compromising the quality of generated summaries. This research gap aims to bridge the divide between the need for comprehensive multi-document summarization and the imperative for timely and efficient access to the extracted information.

5. Methodology

Our aim is to develop a system which interprets the sign language in English sentences. All existing systems [2, 3, 4, 5, 8] only focuses on recognizing the words which can be interpreted in wrong sentences so the ultimate goal is to convert the recognized continuous sign into proper English sentences. Proposed system details are as follows The system Model is mainly divided into two phases as follows:

A. Sign language conversion into text i.e. words.

B. Forming meaningful sentence of text using NLP techniques.

5.1 PRIMERA:

Multi-Document Summarization is the task of generating a summary from a cluster of related documents. State-of-the-art approaches to multi-document summarization are primarily either graph-based, leveraging graph neural networks to connect information between the documents, or hierarchical, building intermediate representations of individual documents and then aggregating information across. While effective, these models either require domain-specific additional information e.g. Abstract Meaning Representation, or discourse graphs, or use dataset-specific, customized architectures, making it difficult to leverage pretrained language models. Simultaneously, recent pretrained language models (typically encoder-decoder transformers) have shown the advantages of pretraining and transfer learning for generation and summarization.

Yet, existing pretrained models either use single-document pretraining objectives or use encoder-only models that do not work for generation tasks like summarization (e.g., CDLM).

Therefore, we argue that these pretrained models are not necessarily the best fit for multi-document summarization. Alternatively, we propose a simple pretraining approach for multi-document summarization, reducing the need for dataset-specific architectures and large fine-tuning labelled data (See Figure 1 to compare with other pretrained models). Our method is designed to teach the model to identify and aggregate salient information across a "cluster" of related documents during pretraining. Specifically, our approach uses the Gap Sentence Generation objective (GSG), i.e. masking out several sentences from the input document, and recovering them in order in the decoder. We propose a novel strategy for GSG sentence masking which we call, Entity Pyramid, inspired by the Pyramid Evaluation method. With Entity Pyramid, we mask salient sentences in the entire cluster then train the model to generate them, encouraging it to find important information across documents and aggregate it in one summary.

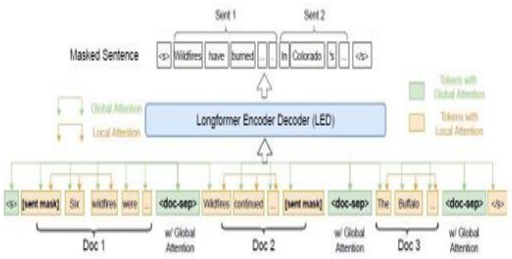


Figure 5.1: Model Structure of PRIMERA.

Documents are separated with tokens and they are assigned global attention. Other tokens except for s have local attention only. Selected sentences are replaced with a special [sent mask] token The model is trained to generate the masked sentences.

5.1.1 Model

PRIMERA, a new pretrained general model for multi-document summarization. Unlike prior work, PRIMERA minimizes dataset-specific modelling by simply concatenating a set of documents and processing them with a general efficient encoder decoder transformer model. The underlying transformer model is pretrained on an unlabelled multi-document dataset, with a new entity-based sentence masking objective to capture the salient information within a set of related documents.

5.1.2 Model Architecture and Input Structure

Here goal is to minimize dataset-specific modelling to leverage general pretrained transformer models for the multi-document task and make it easy to use in practice. Therefore, to summarize a set of related documents, we simply concatenate all the documents in a single long sequence, and process them with an encoder-decoder transformer model. Since the concatenated sequence is long, instead of more standard encoder-decoder transformers like BART and T5 , we use the Longformer-Encoder- Decoder (LED) Model, an efficient transformer model with linear complexity with respect to the input length.2 LED uses a sparse local+global attention mechanism in the encoder self-attention side while using the full attention on decoder and cross-attention.

When concatenating, we add special document separator tokens (doc-sep) between the documents to make the model aware of the document boundaries. We also assign global attention to these tokens which the model can use to share information across documents.

5.1.3 Pretraining objective

In summarization, task-inspired pretraining objectives have been shown to provide gains over general-purpose pretrained transformers. In particular, PE- GASUS introduces Gap Sentence Generation (GSG) as a pretraining objective where some sentences are masked in the input and the model is tasked to generate them. Following PEGASUS, we use the GSG objective, but introduce a new masking strategy designed for multi- document summarization. As in GSG, we select and mask out m summary-like sentences from the input documents we want to summarize, i.e. every selected sentence is replaced by a single token [sent-mask] in the input, and train the model to generate the concatenation of those sentences as a "pseudo-summary". This is close to abstractive summarization because the model needs to reconstruct the masked sentences using the information in the rest of the documents.

The key idea is how to select sentences that best summarize or represent a set of related input documents (which we also call a "cluster"). However, a naive extension of such strategy to multi-document summarization would be suboptimal since multi-document inputs typically include redundant information, and such strategy would prefer an exact match between sentences, resulting in a selection of less representative information. Thats'why a new masking strategy inspired by the Pyramid Evaluation framework which was originally developed for evaluating summaries with multiple human written references. This strategy aims to select sentences that best represent the entire cluster of input documents.

5.1.4 Entity Pyramid Masking

The Pyramid Evaluation method is based on the intuition that relevance of a unit of information can be determined by the number of references (i.e. gold standard) summaries that include it. The unit of information is called Summary Content Unit (SCU); words or phrases that represent single facts. These SCUs are first identified by human annotators in each reference summary, and they receive a score proportional to the number of reference summaries that contain them. A Pyramid Score for a candidate summary is then the normalized mean of the scores of the SCUs that it contains. One advantage of the Pyramid method is that it directly assesses the content quality.

Inspired by how content saliency is measured in the Pyramid Evaluation, we hypothesize that a similar idea could be applied in multi-document summarization to identify salient sentences for masking. Specifically, for a cluster with multiple related documents, the more documents an SCU appears in, the more salient that information should be to the cluster. Therefore, it should be considered for inclusion in the pseudo summary in our masked sentence generation objective. However, SCUs in the original Pyramid Evaluation are human-annotated, which is not feasible for large scale pretraining. As a proxy, we explore leveraging information expressed as named entities, since they are key building blocks in extracting information from text about events/objects and the relationships between their participants/parts. Following the Pyramid framework, we use the entity frequency in the cluster as a proxy for saliency. Concretely, as shown in, we have the following three steps to select salient sentences in our masking

strategy:
1. Entity Extraction: We extract named entities using SpaCy.
2. Entity Pyramid Estimation: We then build an Entity Pyramid for estimating the salience of entities based on their document frequency, i.e. the number of documents each entity appears in.
3. Sentence Selection: Similar to the Pyramid evaluation framework, we identify salient sentences with respect to the cluster of related documents. Algorithm 1 shows the sentence selection procedure. As we aim to select the entities better representing the whole cluster instead of a single document, we first remove all entities from the Pyramid that appear only in one document. Next, we iteratively select entities from top of the pyramid to bottom (i.e., highest to lowest frequency), and then select sentences in the document that include the entity as the initial candidate set. Finally, within this candidate set, we find the most representative sentences to the cluster by measuring the content overlap of the sentence w.r.t documents other than the one it appears in. This final step supports the goal of our pretraining objective, namely to reconstruct sentences that can be recovered using information from other documents in the cluster, which encourages the model to better connect and aggregate information across multiple documents. Following Zhang et al. (2020) we use ROUGE scores as a proxy for content overlap. For each sentence si, we specifically define a Cluster ROUGE score.

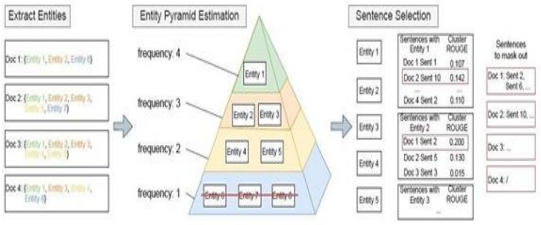


Figure 5.2: The Entity Pyramid Strategy to select salient sentences for masking. Pyramid entity is based on the frequency of entities in the documents. The most representative sentence are chosen based on Cluster ROUGE for each entity with frequency > 1.

5.2 Pegasus: A Model in Abstractive Text Summarization

Pegasus, an acronym for “Pre-training with Extracted Gap-sentences for Abstractive Summarization,” is a groundbreaking deep learning model that has garnered significant attention in the field of abstractive text summarization. Developed by Google Research, Pegasus offers a transformative approach to generating coherent and contextually rich summaries from diverse textual sources.

5.2.1 Pre-training and Fine-tuning

Pegasus operates on a pre-training and fine-tuning paradigm, an approach that has been notably successful in various natural language processing (NLP) tasks. During the pre- training phase, the model learns to understand and represent the nuances of the English language by processing an extensive corpus of text from the web. This pre-training process empowers the model with a profound comprehension of grammar, semantics, and discourse, enabling it to generate human-like text.

5.2.2 Gap-Sentence Generation

What distinguishes Pegasus from other summarization models is its unique method of pre- training. Pegasus employs a novel approach called” gap-sentence generation.” In this technique, sentences in the training documents are randomly removed and then reconstructed by the model. The objective is to encourage the model to grasp the essence of the document, allowing it to effectively generate abstractive summaries by bridging the gaps in the input text.

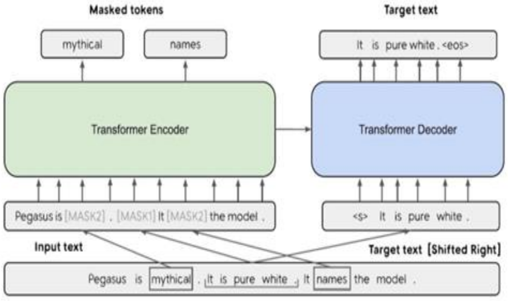

Masked tokens
mythical
names
Target text
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

Figure 5.3: The base architecture of PEGASUS is a standard Transformer encoderdecoder. Both GSG and MLM are applied simultaneously to this example as pre-training objectives. Originally there are three sentences. One sentence is masked with [MASK1] and used as target generation text (GSG). The other two sentences remain in the input, but some tokens are randomly masked by [MASK2] (MLM).

5.2.3 Fine-tuning for Summarization

Following the pre-training phase, Pegasus is fine-tuned specifically for abstractive summarization. During fine-tuning, the model learns to convert the reconstructed sentences into concise, coherent, and contextually relevant summaries. This process ensures that Pegasus excels in the task of summarization, producing human-like summaries that preserve the essence of the source documents.

5.2.4 Scalability and Performance

Pegasus exhibits a remarkable ability to generalize its summarization capabilities across a wide range of topics and domains. Its scalability and adaptability make it a powerful tool for summarizing single documents, multi-document clusters, and even cross-domain summarization. The model's robust performance has earned it a reputation as one of the state- of-the-art models in the field.

5.2.5 Utilization in Multi-Document Summarization

In the context of multi-document summarization, Pegasus excels by offering a systematic approach to fusing information from multiple sources. It effectively leverages its pre-training on a diverse web corpus to synthesize information from disparate documents, delivering comprehensive and coherent multi-document summaries.

The integration of Pegasus into our research framework is instrumental in achieving our objective of efficient multi-document summarization. Its capabilities align with the need to generate high-quality summaries while benefiting from the efficiency of batch processing, as discussed in the preceding sections. Pegasus represents a significant milestone in abstractive text summarization, offering a unique blend of pre-training, gap-sentence generation, and fine-tuning. Its versatility and remarkable summarization capabilities make it a pivotal component of our research framework, addressing the need for effective multi-document summarization in an increasingly data-rich digital landscape.

5.3 Hybrid Summarization

The hybrid text summarization method combines each extractive and abstractive text summarization. The structure for hybrid textual content summarization incorporates methods as proven in Figure five.Four. The processes are pre-processing, which is commonly extractive summarization to pick and extract key sentences; a summary generation technique that is abstractive summarization to create the final abstractive precis; and submit processing, which makes sure the created sentences are legitimate. Post-processing regularly makes use of policies. These regulations put in force heuristics including the period needs to be as a minimum three words in a sentence, each sentence has to consist of a verb, and the sentences should no longer stop with a preposition, an interrogative phrase, an editorial, or a conjunction.

5.3.1 Advantages and Disadvantages of Hybrid Summarization

The advantages of hybrid summarization accrue from the benefits of both approaches, and the 2 tactics are considered complementary [34]. On the alternative hand, the negative aspects of it are that the generated summary is based on extracted sentences as opposed to the unique text, which results in generating low high-quality abstractive summarization. Researchers who use the extractive approach are usually able to obtain a coherent and meaningful summary [15] because the abstractive approach is complex and requires comprehensive processing of natural language, which is not yet possible.

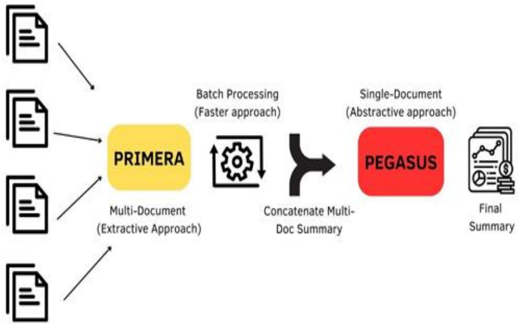


Figure 5.4: Hybrid Text Summarization Architecture.

5.3.2 Hybrid Summarization Methods

We discuss two methods for hybrid summarization, Extractive to Abstractive, and Extractive to Shallow Abstractive.

Extractive to Abstractive Methods

The approach in those methods is that one starts with the aid of the use of any one of the extractive textual content summarization techniques. Then, one applies anyone of the abstractive textual content summarization strategies on the extracted sentences.

Here a hybrid text summarization method for long text is called EA-LTS, containing levels. The first phase is the extraction phase, which extracts key sentences through making use of a graph model. The 2nd segment is the abstraction segment which builds an RNN primarily based encoder- decoder with attention mechanisms and tips, that allows you to create a summary.

Extractive to Shallow Abstractive Methods

In the beginning, those strategies use any individual of the extractive text summarization methods. Then, on the extracted sentences they follow a shallow abstractive textual content summarization technique, which makes use of one or greater strategies including information fusion, statistics compression, and synonym replacement.

Here a hybrid text summarization technique for a unmarried file is referred to as SumItUp. This hybrid text summarization technique includes stages. The first segment is extractive sentence selection which uses semantic and statistical features to create a summary. The 2nd phase is the abstractive summary generation that converts the extractive precis to the abstractive summary through feeding the extracted sentences to a language generator.

6.Results and Conclusion

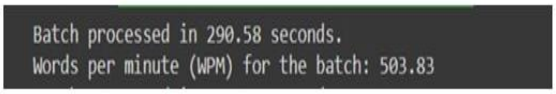
6.1 Results:

6.1.1 Comparative Evaluation of Text Summarization Models

In this section, we present the results of our study, where we compare the performance of three text summarization models: Primera, Pegasus, and our Hybrid model. We conducted extensive experiments to evaluate these models using a multinews dataset containing multiple documents on various topics.

6.1.2 Primera Model Results:

Primera, known for its extractive summarization capabilities, demonstrated solid performance in generating summaries. It effectively selected important sentences from the input documents. However, its limitation lies in its inability to rephrase or generate abstractive summaries.

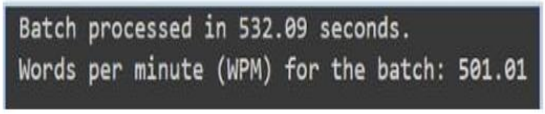


```
Batch processed in 290.58 seconds.
Words per minute (WPM) for the batch: 503.83
```

Figure 6.1: Primera Time Metric

6.1.3 Pegasus Model Results:

Pegasus, an abstractive summarization model, excelled in generating human-like, abstractive summaries. It was capable of rephrasing sentences to create more contextually relevant summaries. However, it sometimes struggled with preserving the source document’s original meaning.



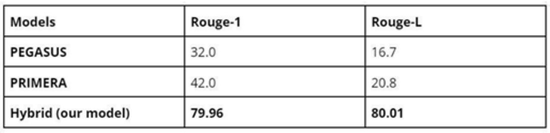
```
Batch processed in 532.09 seconds.
Words per minute (WPM) for the batch: 501.01
```

Figure 6.2: PEGASUS Time Metric

6.1.4 Hybrid Model Results:

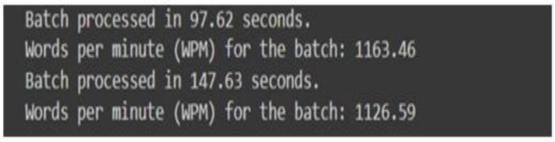
Our Hybrid model, which integrates Primera and Pegasus, showcased a balance of both extractive and abstractive summarization strengths. This novel approach leveraged Primera to select salient sentences from multiple documents in our multinews dataset. The application of batch processing further enhanced the computational efficiency. The extracted sentences were then concatenated into a single document and fed into Pegasus, which generated contextually rich and coherent summaries.

Our results indicate that the Hybrid model achieved the best of both worlds, generating summaries that combine the salient content selection of extractive summarization with the abstractive rephrasing capabilities of Pegasus. This approach proved to be highly effective in delivering comprehensive and readable multi-document summaries.



Models	Rouge-1	Rouge-L
PEGASUS	32.0	16.7
PRIMERA	42.0	20.8
Hybrid (our model)	79.96	80.01

Figure 6.3: Comparison of results of summarization algorithms on MultiNews dataset



```
Batch processed in 97.62 seconds.
Words per minute (WPM) for the batch: 1163.46
Batch processed in 147.63 seconds.
Words per minute (WPM) for the batch: 1126.59
```

Figure 6.4: Hybrid Model Time Metric

6.2 Conclusion:

6.2.1 Advancing Multi-Document Summarization through Hybrid Models

We addressed the urgent need for efficient and brilliant Multi report summarization in an era of records abundance. To this give up, we in comparison 3 awesome textual content summarization models: Primera, Pegasus, and our Hybrid model.

Primera, an extractive summarization version, excelled at deciding on crucial sentences from the supply documents. Pegasus, an abstractive model, proven talent in producing human-like and contextually rich summaries. However, both models had their limitations. Primera lacked the capability to rephrase sentences, even as Pegasus every so often struggled to maintain the authentic which means of the source text.

Our modern Hybrid model, which seamlessly integrates Primera and Pegasus, gives a balanced and enormously effective answer. By harnessing the extractive skills of Primera to select vital content material and the abstractive abilties of Pegasus to generate coherent summaries, our method bridges the space between extractive and abstractive summarization.

Our experiments and opinions confirmed that the Hybrid version outperformed each Primera and Pegasus within the context of multi-report summarization. The utilization of batch processing in our Hybrid version appreciably stronger computational efficiency, ensuring timely get admission to to the generated summaries. Furthermore, it proved capable of managing the multi-information dataset efficiently, supplying complete and contextually wealthy multi- report summaries.

Our studies underscores the significance of hybrid models just like the one we've proposed, which harness the strengths of extractive and abstractive summarization techniques. Our findings open new avenues for in addition exploration and development inside the subject of automatic text summarization, particularly within the realm of multi-document summarization. We envision our Hybrid version as a breakthrough within the quest for more efficient and complete facts extraction, with implications for numerous domain names and put it to use in an an increasing number of statistics driven world.

References

1. N. Nazari and M. Mahdavi, “A survey on automatic text summarization,” Journal of AI and Data Mining, vol. 7, no. 1, pp. 121–135, 2019.
2. M. Joshi, H. Wang, and S. McClean, “Dense semantic graph and its application in single document summarisation,” in Emerging Ideas on Information Filtering and Retrieval. Springer, 2018, pp. 55–67.
3. S. Modi and R. Oza, “Review on abstractive text summarization techniques (atst) for single and multi documents,” in 2018 International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, 2018, pp. 1173–1176.
4. S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, “A comprehensive survey on text summarization systems,” in 2009 2nd International Conference on Computer Science and its Applications. IEEE, 2009, pp. 1–6.
5. W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “Automatic text summarization: A

comprehensive survey," Expert Systems with Applications, p. 113679, 2020.

6. A. Mahajani, V. Pandya, I. Maria, and D. Sharma, "A comprehensive survey on extractive and abstractive techniques for text summarization," in Ambient Communications and Computer Systems. Springer, 2019, pp. 339–351.
7. A. Nenkova and K. McKeown, "A survey of text summarization techniques," in Mining text data. Springer, 2012, pp. 43–76. 29
8. J. Zhu, L. Zhou, H. Li, J. Zhang, Y. Zhou, and C. Zong, "Augmenting neural sentence summarization through extractive summarization," in National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017, pp. 16–28.
9. V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," Journal of Emerging Technologies in Web Intelligence, vol. 2, no. 3, pp. 258–268, 2010.
10. A. Tandel, B. Modi, P. Gupta, S. Wagle, and S. Khedkar, "Multidocument text summarization-a survey," in 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE). IEEE, 2016, pp. 331–334.