EUROPEAN JOURNAL OF EMERGING INTELLIGENT AUTOMATION AND CONTROL SYSTEMS

LEVERAGING ENCODER-DECODER TRANSFORMER MODELS FOR ABSTRACTIVE TEXT SUMMARIZATION OF SOCIAL MEDIA CONTENT

Dr. Katarina L. Novak
Department of Computer Science, University of Zagreb, Croatia

Dr. Mika S. Kato
Graduate School of Information Science, Nagoya University, Japan

Dr. Elena V. Smirnova
Department of Computational Linguistics, Saint Petersburg State University, Russia

V0LUME01 ISSUE01 (2024)

Published Date: 19 December 2024 // Page no.: - 48-66

ABSTRACT

The pervasive nature of social media platforms has led to an overwhelming volume of textual data, necessitating efficient methods for information extraction and summarization. This article explores the application of encoder-decoder transformer models for abstractive text summarization of social media content. Traditional summarization techniques often struggle with the inherent brevity, informality, and noise of social media data. We delve into the foundational principles of transformer architectures, including their encoder-decoder framework and attention mechanisms, which enable them to generate coherent and contextually relevant summaries. The article discusses prominent pre-trained models such as BART, T5, and PEGASUS, highlighting their specific pre-training objectives that make them suitable for abstractive summarization. Furthermore, it touches upon the emerging role of prompt engineering in guiding these models. We analyze the demonstrated effectiveness of transformer models in handling the unique challenges of social media summarization, including their capacity for abstractive generation and robustness to noisy input, as evidenced by evaluation metrics like ROUGE. Finally, the discussion addresses current limitations such as real-time processing, multimodal content integration, and factual consistency, while proposing future research directions, including enhanced domain adaptation, handling long social media aggregations, and leveraging Large Language Models (LLMs). This work underscores the critical role of advanced abstractive summarization in navigating the complexities of social media information.

Keywords: Abstractive Summarization, Social Media, Encoder-Decoder Models, Transformers, Natural Language Processing, Deep Learning, BART, T5, PEGASUS, Prompt Engineering.

INTRODUCTION

The dawn of the digital age, propelled by rapid advancements in internet and smart technology, has fundamentally reshaped human communication and information consumption. Social media platforms, in particular, have ascended to a dominant position, serving as vibrant ecosystems for information dissemination, advertising, opinion exchange, and emotional expression. The sheer volume of posts, comments, and interactions generated daily on platforms like Facebook, Twitter (now X), Reddit, and others creates an unprecedented wealth of data. These digital footprints collectively shape public opinion on myriad issues, ranging from political discourse and economic trends to social movements and personal experiences, cementing social media's pivotal role in contemporary public life [11, 24, 36].

The ability of individuals to express themselves with a degree of anonymity and a reduced sense of direct interpersonal confrontation on social networks often leads to more candid and representative opinions compared to traditional face-to-face interactions [24, 36]. This candidness transforms social media data into a cornerstone for research, offering invaluable insights into patterns of social behavior. Such insights can inform critical decisions across governmental policies, business strategies, and broader societal initiatives. However, this deluge of information presents a formidable challenge: when a significant event unfolds, social networks become inundated with posts and comments, rendering it practically impossible for any individual to consume all relevant content manually [24, 36]. Consequently, the development of automated systems capable of generating concise and coherent summaries of social media content has become not merely desirable, but indispensable.

Automatic text summarization, a long-standing area within Natural Language Processing (NLP), aims to distil the essence of a source text into a shorter, more digestible version while preserving its core meaning and crucial information [1, 34]. Broadly, summarization techniques are categorized into two main types: extractive and abstractive [1, 37]. Extractive summarization operates by identifying and extracting the most significant sentences or phrases directly from the original document and concatenating them to form a summary [4, 34]. While straightforward and often computationally less intensive, extractive summaries can suffer from a lack of fluidity and coherence, as they do not involve linguistic generation or rephrasing. In stark contrast, abstractive summarization transcends mere extraction. It involves a deeper understanding of the source text's meaning, enabling the system to generate novel sentences and phrases that convey the original information concisely, much like a human summarizer would [1, 3, 37]. This generative capability allows abstractive methods to produce more fluent, cohesive, and human-readable summaries, often achieving a higher degree of conciseness and avoiding redundancy [16].

The application of text summarization to the unique characteristics of social media content, however, introduces a distinct set of complexities. Social media posts are typically characterized by their brevity, informality, and often unstructured nature. They frequently incorporate non-standard linguistic elements such as slang, abbreviations, hashtags, emojis, and idiosyncratic grammar [6, 7, 14, 15, 20, 22, 23, 24, 25, 26, 33, 36]. These peculiarities render traditional summarization approaches, typically designed for formal, well-structured documents, less effective. Moreover, the dynamic and real-time nature of social media streams, particularly during unfolding events, demands summarization systems capable of rapid adaptation and continuous updating [11, 12, 23, 26].

Recent years have witnessed a transformative shift in NLP, largely propelled by the emergence of deep learning techniques, most notably the transformer architecture. Introduced by Vaswani et al. in 2017 [5], transformers revolutionized sequence modeling by replacing traditional recurrent or convolutional layers with self-attention mechanisms. This architectural innovation enabled models to process input sequences in parallel, capture long-range dependencies more effectively, and achieve state-of-the-art results across a multitude of NLP tasks, including machine translation, text generation, and, critically, text summarization [17, 18, 19, 21, 38, 39, 44]. Encoder-decoder transformer models, in particular, have demonstrated an exceptional capacity to understand complex textual inputs and generate highly coherent and contextually relevant outputs, making them uniquely suited for abstractive summarization.

This article aims to provide a comprehensive exploration of the application of encoder-decoder transformer models for abstractive text summarization of social media content. We will begin by tracing the historical trajectory of automatic summarization, highlighting the evolution from early statistical methods to advanced neural network approaches. Subsequently, we will delve into the intricate architecture of transformer models, elucidating the role of their key components such as multi-head attention and positional encodings. A significant portion will be dedicated to discussing prominent pre-trained encoder-decoder models (e.g., BART, T5, PEGASUS) that have become benchmarks in abstractive summarization, alongside the contemporary relevance of prompt engineering. The paper will then detail a methodological framework for applying these models to social media data, covering aspects from data collection and preprocessing to topic-based grouping and summary generation. We will present illustrative results and discuss the nuances of evaluating summarization quality using metrics like ROUGE. Finally, the discussion will critically examine the current limitations and ongoing challenges in social media summarization using transformer models, while concurrently outlining promising avenues for future research and development in this dynamic field.

2. Text Summarization Methods: A Comprehensive Overview

Automatic text summarization is a critical subfield of Natural Language Processing (NLP) dedicated to condensing large volumes of text into shorter, coherent, and informative summaries. In an era of exponential information growth, particularly amplified by the digital landscape and social media, effective summarization tools are indispensable for managing information overload and facilitating rapid comprehension [33, 34]. The ultimate goal is to enable machines to generate summaries that rival the quality and nuance of those produced by humans, preserving key information and the overall meaning of the original content.

Historically, the pursuit of automated summarization dates back over half a century. Early efforts, such as those by Luhn in 1958, focused on statistical methods to identify and extract important sentences based on criteria like word frequency [2]. Over the decades, the field has evolved considerably, giving rise to distinct approaches, each with its unique mechanisms, advantages, and limitations.

There are three primary categories of text summarization approaches: extractive, abstractive, and hybrid [34]. This section will provide a detailed exposition of each approach, exploring their underlying methods and their particular relevance to the complexities of social media content.

2.1. Extractive Summarization

Extractive summarization is the more straightforward of the two main approaches. In this paradigm, the summarization system operates by identifying and directly extracting key sentences, phrases, or clauses from the original source text. These extracted segments are then concatenated, with minimal alteration, to form the final summary. The core principle is akin to highlighting important passages in a document.

The process of extractive summarization typically involves a series of sequential steps:

1. Text Representation and Analysis: The initial step involves converting the input text into a suitable numerical or structural representation that facilitates analysis. This often includes tokenization (breaking text into words/sentences), part-of-speech tagging, and dependency parsing to understand grammatical relationships. The goal is to prepare the text for identifying significant content units.

2. Content Unit Scoring: Once the text is represented, each potential content unit (typically a sentence, but sometimes phrases or paragraphs) is assigned a score based on various criteria. These scoring mechanisms often rely on statistical, linguistic, or heuristic features. Common features include:

○ Term Frequency-Inverse Document Frequency (TF-IDF): Sentences containing words that are frequent in the current document but rare across a larger corpus are often considered more important.

○ Sentence Position: Sentences at the beginning or end of a document (or paragraph) are often assumed to contain key information.

○ Keywords/Keyphrases: Sentences containing a high density of pre-defined keywords or automatically extracted keyphrases are highly scored.

○ Sentence Length: Optimal sentence length can be a factor, avoiding overly short or long sentences.

○ Lexical Chains: Identifying sequences of related words can help pinpoint central themes and, consequently, important sentences.

○ Graph-based Ranking: Algorithms like TextRank or LexRank build a graph where nodes are sentences and edges represent similarity. Sentences with higher "centrality" (more connections to other similar sentences) are ranked higher.

3. Summary Generation (Extraction): After scoring, a selection mechanism is employed to choose the top-scoring sentences (or a fixed percentage of the highest-scoring sentences) until a desired summary length is achieved. A crucial aspect here is the length constraint, which is managed through mechanisms like word count limits, sentence count limits, or thresholding on scores [35]. Importantly, the selected sentences are typically presented in their original order to maintain some semblance of coherence.

Advantages of Extractive Summarization:

● Simplicity and Speed: Extractive methods are generally simpler to implement and computationally less demanding compared to abstractive approaches.

● Accuracy and Factuality: Since the summary consists of direct excerpts from the original text, it inherently maintains the factual accuracy and terminology of the source. This reduces the risk of "hallucinations" or inaccuracies, a problem often associated with generative models.

● Traceability: It's easy to trace back the source of any sentence in the summary to the original document, which can be important for verification.

● Domain Agnostic: Many extractive techniques are less dependent on domain-specific knowledge or extensive training data, making them broadly applicable.

Disadvantages of Extractive Summarization:

● Lack of Coherence and Fluency: The primary drawback is that concatenating extracted sentences, even highly relevant ones, often results in a summary that lacks grammatical cohesion, smooth transitions, and overall fluency. There might be abrupt topic shifts between sentences.

● Redundancy: Extractive methods may include redundant information if similar concepts are expressed in different ways across multiple extracted sentences. Without semantic understanding, the system cannot easily identify and eliminate such repetitions.

● Grammatical Incorrectness: While individual extracted sentences are grammatically correct, their concatenation might lead to awkward phrasing or a disjointed narrative.

● Limited Compression: The degree of compression achievable is limited by the granularity of extraction (typically sentences). This can lead to longer summaries than necessary, especially when only a small piece of information within a sentence is truly relevant.

● Inflexibility: Extractive summaries cannot rephrase, generalize, or synthesize information in a human-like manner, which limits their ability to truly "understand" and express the core meaning of the document [37].

Despite these limitations, extractive summarization remains relevant, particularly in scenarios where high fidelity to the original text and computational efficiency are paramount.

2.2. Abstractive Summarization

Abstractive summarization represents a more advanced and complex approach, aiming to generate a summary that is not merely a collection of extracted sentences but a novel piece of text. This approach mimics human summarization, where individuals read and comprehend a document, synthesize its main ideas, and then rephrase them in new words and sentences to create a concise, fluent, and grammatically correct summary [1, 3, 37]. The generated summary may not contain any direct sentences from the original text, making it appear entirely new while retaining the original meaning [36].

The abstractive summarization process fundamentally involves two sophisticated tasks:

1. Semantic Understanding and Representation: The summarizer must first deeply comprehend the input text to build an internal semantic representation. This involves identifying key concepts, entities, relationships, events, and the overall discourse structure. Unlike extractive methods that focus on surface-level features, abstractive systems delve into the meaning of the text to identify core information. This often requires advanced NLP techniques, including named entity recognition, coreference resolution, and semantic role labeling.

2. Natural Language Generation (NLG): Once the semantic representation is formed, the system employs NLG techniques to synthesize new sentences that convey the extracted meaning concisely. This involves selecting appropriate vocabulary, constructing grammatically correct sentences, ensuring logical flow, and maintaining stylistic consistency. This generative aspect is what sets abstractive summarization apart, demanding capabilities similar to those found in machine translation or dialogue generation systems.

Early abstractive approaches often relied on rule-based systems, semantic parsing, or template-based generation. However, these methods were highly complex to design, brittle, and lacked scalability. The paradigm shift occurred with the advent of deep learning and neural networks, particularly the encoder-decoder architectures, which proved capable of learning the complex mapping from input text to summary output.

Advantages of Abstractive Summarization:

- Human-like Quality: Summaries generated abstractively are generally more fluent, coherent, and readable than extractive ones, closely resembling summaries produced by humans. They can paraphrase, generalize, compress, or fuse information effectively.
- Higher Compression Ratios: By rephrasing and synthesizing, abstractive models can achieve higher compression ratios, producing much shorter summaries that still convey the essential information, without being constrained by sentence boundaries of the original text.
- Redundancy Avoidance: Since the model generates new text based on its understanding, it can inherently avoid repeating information that might be present in different forms across the source document.
- Improved Clarity: Abstractive summaries can rephrase complex or ambiguous original sentences into clearer, more concise language, improving overall readability.
- Addressing Information Scatter: They can gather information scattered across different parts of the original text and present it cohesively in a single summary statement.

Disadvantages of Abstractive Summarization:

- Complexity of Implementation: Abstractive summarization is significantly more challenging to develop and implement. It requires sophisticated models capable of deep language understanding and fluent generation, which is still an active area of research [37].
- Risk of Hallucination: A major concern is the potential for "hallucination," where the model generates content that is plausible but factually incorrect or not supported by the source text [1]. This is a critical issue, especially in domains requiring high factual accuracy like news or scientific articles.
- Requires Large Datasets: Training abstractive models, especially deep neural networks like transformers, demands vast amounts of high-quality, paired (document, summary) datasets, which are often expensive and time-consuming to create.
- Computational Expense: These models are typically larger and require significant computational resources for training and inference.
- Lack of Traceability: Because new sentences are generated, it can be difficult to trace the specific source phrases in the original text that contributed to a particular part of the summary, making verification challenging.
- Handling Out-of-Vocabulary (OOV) Words: Dealing with words not encountered during training (OOV words), especially in dynamic domains like social media, can be problematic.

Despite these challenges, the superior quality and flexibility of abstractive summaries make them the preferred choice for many applications, especially with the advancements in transformer-based models.

2.3. Hybrid Summarization

Hybrid summarization approaches attempt to combine the strengths of both extractive and abstractive techniques while mitigating their individual weaknesses. The fundamental idea is to leverage the robustness and factual grounding of extractive methods with the fluency and conciseness of abstractive generation.

A typical hybrid summarization process often involves a two-stage procedure:

1. Extractive Pre-processing/Filtering: In the first stage, an extractive component is used to identify and extract the most salient sentences or phrases from the original document. This step acts as a filtering mechanism, reducing the amount of input text that the abstractive model needs to process. By focusing on highly relevant segments, this stage helps to improve the efficiency and accuracy of the subsequent abstractive generation. This extraction can be based on various features, similar to pure extractive methods.

2. Abstractive Refinement/Generation: The extracted content, which is already a condensed version of the original, then serves as the input for an abstractive summarizer. This abstractive component rephrases, synthesizes, and condenses the extracted sentences into a

more fluent, coherent, and human-like summary. The abstractive model's task is simplified because it operates on a smaller, pre-filtered set of highly relevant sentences rather than the entire verbose original document.

Advantages of Hybrid Summarization:

- Combines Strengths: It aims to achieve a balance between factual accuracy (from extraction) and linguistic quality (from abstraction).
- Reduced Complexity for Abstraction: By providing a pre-filtered input, the abstractive component's task becomes less complex, potentially leading to better performance and reduced computational demands compared to pure abstractive methods on very long texts.
- Improved Factual Grounding: The initial extractive step can help to ground the abstractive output more firmly in the source text, potentially reducing hallucination issues.
- Scalability for Long Documents: This approach can be particularly beneficial for very long documents, where directly applying an abstractive model might be computationally prohibitive or lead to information loss.

Disadvantages of Hybrid Summarization:

- Cascading Errors: Errors made in the extractive stage (e.g., extracting irrelevant sentences or missing crucial ones) can propagate and negatively impact the quality of the final abstractive summary.
- Increased System Complexity: The system architecture becomes more complex due to the integration of two distinct summarization paradigms.
- Sub-optimal Abstractive Quality: The final summary might not be as "purely" abstractive as a summary generated by a full-fledged abstractive model operating on the entire original text, as its creativity is constrained by the extracted sentences. The PDF suggests that the generated summary might not be as qualitative as a purely abstractive summary because it's based on extracted key sentences rather than the original text as a whole.

Hybrid approaches are increasingly gaining traction, especially for summarizing lengthy documents or when a balance between factual correctness and linguistic quality is desired. They represent a pragmatic middle ground in the evolving landscape of text summarization.

2.4. Social Media Summarization Significance and Challenges

The significance of social media summarization stems directly from the unparalleled role these platforms play in contemporary life. Social media is not merely a communication tool; it is a dynamic, real-time reflection of collective consciousness, a primary conduit for news, political discourse, economic trends, and societal sentiment. The posts and comments exchanged on these platforms are potent shapers of public opinion, influencing everything from governmental policy-making to consumer behavior and social research [24, 36].

The informal and often unfiltered nature of social media interactions means that opinions expressed online are frequently more direct and representative of genuine public sentiment than those in offline, face-to-face communications [24, 36]. This candidness is partly attributed to the perceived anonymity and privacy of online interactions, which can lower inhibitions and reduce the impact of social stereotypes. Consequently, data circulating on social media is a rich, often unparalleled, source for extracting patterns of social behavior and underlying emotions. While statistical and graphical representations can quantify sentiments (e.g., a movie rating from 0-5), a text summary offers a qualitative, coherent overview – detailing a movie's theme, a product's performance issues, or nuances of a political debate [36].

The imperative for effective social media summarization is amplified during critical events. When a major incident breaks, social networks are instantly deluged with an overwhelming number of posts and comments. The sheer volume makes manual consumption impractical, and the often redundant or repetitive nature of these contributions can lead to confusion and information overload for readers [24, 36]. Therefore, creating accurate, timely, and concise summaries of this chaotic information stream is crucial for enabling individuals and organizations to stay informed and make sense of rapidly unfolding events.

However, the unique characteristics of social media content present a formidable array of challenges for traditional text summarization techniques:

1. Informality and Ill-formed Language: Unlike formal documents or news articles, social media content is inherently informal. It often deviates significantly from standard grammatical rules, sentence structures, and conventional lexicon. Users frequently employ colloquialisms, slang, dialectal variations, and highly condensed expressions [6, 7, 24, 36]. This "noisy" language can pose significant hurdles for NLP models trained predominantly on formal text.
2. Abbreviation, Special Characters, and Emojis: Social media communication is replete with platform-specific shorthand. This includes widespread use of abbreviations (e.g., "LOL," "BRB"), acronyms, emoticons, emojis, and special characters (e.g., hashtags #topic, mentions @user, URLs). These elements are integral to conveying meaning and sentiment but are often challenging for summarization models to interpret and integrate effectively into coherent summaries.
3. Lack of Lexical Richness and Brevity: Individual social media posts, particularly tweets or short comments, are characterized by extreme brevity. This leads to a limited lexical richness within single entries. While a

collection of tweets on a topic might be rich in information, each individual unit is typically concise and fragmented [24, 36]. Summarization models must therefore learn to synthesize information from numerous short, lexically sparse inputs to form a comprehensive output.

4. Dynamic and Real-time Nature: Social media content is constantly evolving. During an event, posts can emerge and change rapidly. Summarization systems need to be capable of processing these dynamic streams in real-time, adapting their summaries as new information or shifts in sentiment occur. This real-time requirement imposes significant computational and algorithmic demands [11, 12, 23, 26].

5. Multimodality: Modern social media is increasingly multimodal, integrating text with images, videos, GIFs, and links [8, 9, 10]. A truly comprehensive summary of social media content might need to consider information embedded in these non-textual modalities. Current transformer models are primarily text-centric, making multimodal summarization a complex, yet critical, future direction [10].

6. Redundancy and Repetition: Despite their brevity, social media streams often contain a high degree of redundancy and repetitive information, as many users might react similarly to an event or share the same news. Effective summarization systems must be adept at identifying and eliminating this repetition to produce truly concise summaries [24].

7. Subjectivity and Sentiment: Social media is a hotbed for expressing subjective opinions, emotions, and sentiments. Summarization systems need to be able to capture and reflect the prevailing sentiment or diverse viewpoints within a discussion without introducing bias or misrepresentation.

These unique characteristics make social media summarization a challenging but incredibly valuable task. The inherent difficulties highlight why traditional methods are often insufficient and why advanced approaches, particularly transformer-based abstractive summarization techniques, are increasingly becoming the focus of research. These models, with their sophisticated attention mechanisms and generative capabilities, show promise in managing the peculiarities of social media content to deliver informative and coherent summaries.

3. Transformer Architecture and Methodology for Summarization

The remarkable progress in Natural Language Processing over the past few years can largely be attributed to the advent and widespread adoption of transformer models. These architectures have fundamentally changed how sequential data, particularly text, is processed and understood. This section will delve into the core of the transformer model, explaining its components and how it operates, before discussing the methodology of utilizing pre-trained transformer models for abstractive summarization, specifically in the context of social media content.

3.1. The Transformer Model: "Attention Is All You Need"

The transformer model, introduced by Vaswani et al. in their seminal 2017 paper "Attention Is All You Need" [5], marked a significant departure from previous sequence-to-sequence (Seq2Seq) architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). While RNNs (including LSTMs and GRUs) process sequences sequentially, making them slow and prone to vanishing/exploding gradients over long dependencies, and CNNs capture local features, the transformer revolutionized this by entirely replacing recurrence and convolution with attention mechanisms [5, 38].

The core reasons for the transformer's dominance in NLP are:

1. Exceptional Performance: Transformers have consistently achieved state-of-the-art results across a broad spectrum of NLP tasks, including machine translation, text generation, question answering, sentiment analysis, and, critically, text summarization [17, 18, 19, 21, 44]. Their ability to capture intricate linguistic patterns and semantic relationships is unparalleled.

2. Parallel Processing Capability: Unlike RNNs, which inherently process words one after another (sequentially), transformers leverage attention to allow for simultaneous computation across all words in a sequence. This parallelization significantly reduces training time and computational resource requirements, making it feasible to train much larger models on vast datasets [5].

3. Superior Long-Range Dependency Handling: The self-attention mechanism enables the model to weigh the importance of any word in the input sequence when processing another word, regardless of their distance. This global perspective effectively solves the long-standing problem of capturing long-range dependencies that plagued earlier RNN-based models [5].

The fundamental architecture of the transformer is based on an encoder-decoder framework, a common paradigm for Seq2Seq tasks. This architecture is particularly well-suited for problems like summarization, where an input sequence (the source text) needs to be mapped to an output sequence (the summary) of potentially different lengths and structures.

3.1.1. Encoder Component

The encoder's primary responsibility is to transform the input sequence into a rich, high-dimensional contextualized representation. This representation encapsulates the semantic meaning and syntactic structure of the entire input text. The encoder is typically composed of a stack of Nx identical layers (e.g., Nx=6 in the original paper [5]). Each of these layers contains two main sub-layers:

1. Multi-Head Self-Attention Mechanism: This is the heart of the transformer. For each word in the input sequence, the self-attention mechanism allows the model to compute a weighted sum of all other words in the same sequence. The weights are dynamically determined by the relationships between the words.

○ Query (Q), Key (K), Value (V) Vectors: For each input token, three linear transformations generate Query, Key, and Value vectors. Conceptually, the Query vector represents what information the current word is looking for, the Key vector represents what information the word offers, and the Value vector contains the actual information itself.

○ Scaled Dot-Product Attention: The attention score between a Query and all Keys is computed using a dot product. This score is then scaled by the square root of the dimension of the Key vectors to prevent large values from pushing the softmax function into regions with extremely small gradients. A softmax function is then applied to these scaled scores to obtain probability-like weights. These weights are then multiplied by the Value vectors and summed to produce the output for that head.

○ Multi-Head Attention: Instead of performing a single attention function, Multi-Head Attention linearly projects the Q, K, and V vectors multiple times with different, learned linear projections. This allows the model to attend to different parts of the input from various "representation subspaces" and capture different types of relationships (e.g., syntactic dependencies, semantic similarities) simultaneously. The outputs from these multiple attention "heads" are then concatenated and linearly transformed to produce the final output of the multi-head attention layer [5].

2. Position-wise Fully Connected Feed-Forward Network: This sub-layer consists of two linear transformations with a ReLU activation in between. It is applied independently and identically to each position in the sequence. While simple, it adds non-linearity and allows the model to process the contextualized information further.

Positional Encoding: Since the self-attention mechanism itself is permutation-invariant (meaning it doesn't inherently understand the order of words), the transformer incorporates "positional encodings" to inject information about the relative or absolute position of tokens in the sequence. These encodings are sinusoidal functions (or learned embeddings) that are added to the input embeddings at the bottom of the encoder and decoder stacks. This ensures that the model can leverage word order without relying on recurrence [5].

Residual Connections and Layer Normalization: Around each of the two sub-layers (multi-head attention and feed-forward network), there is a residual connection, followed by layer normalization. Residual connections help in training very deep networks by allowing gradients to flow directly through the network, preventing vanishing gradients. Layer normalization normalizes the inputs across the features, which helps stabilize and speed up training [5].

3.1.2. Decoder Component

The decoder component, also a stack of Nx identical layers, is responsible for generating the output sequence (the summary) based on the contextualized representation provided by the encoder. Each decoder layer has three main sub-layers:

1. Masked Multi-Head Self-Attention: Similar to the encoder's self-attention, but with a crucial modification: it's "masked." This masking ensures that predictions for a given output position can only depend on known outputs (i.e., previous words generated in the summary). This prevents the decoder from "cheating" by looking at future words during training, mimicking the auto-regressive nature of sequence generation.

2. Multi-Head Cross-Attention (Encoder-Decoder Attention): This is a unique and critical layer in the decoder. It performs attention over the output of the encoder stack. The Queries come from the previous decoder sub-layer, while the Keys and Values come from the output of the encoder. This mechanism allows the decoder to "attend" to relevant parts of the source input sequence when generating each word of the summary, effectively linking the generated summary to the original document's content [5].

3. Position-wise Fully Connected Feed-Forward Network: Identical to the one in the encoder, this network processes the combined information from the self-attention and cross-attention layers.

Like the encoder, the decoder layers also employ residual connections and layer normalization. The final layer of the decoder is typically a linear layer followed by a softmax function, which outputs a probability distribution over the model's vocabulary, indicating the likelihood of each word being the next token in the generated summary. The word with the highest probability (or sampled based on strategies like beam search) is selected, and this process continues until an end-of-sequence token is generated or a maximum length is reached.

The entire transformer architecture, by leveraging attention over recurrence, achieves unparalleled efficiency and effectiveness in handling sequential data, making it the bedrock for modern NLP advancements, especially in text generation and summarization.

3.2. Pre-trained Transformer Models and Pipelines

Building transformer models from scratch requires immense computational resources, vast amounts of data, and significant expertise. This prohibitive cost led to the paradigm of pre-trained language models (PLMs), where large transformer models are pre-trained on massive text corpora (e.g., billions of words from books, articles, web pages) to learn general language understanding and generation capabilities. These pre-trained models can then be fine-tuned on smaller, task-specific datasets to achieve state-of-the-art performance with significantly less effort and data [20].

The advantages of using pre-trained transformer models (PTLMs) for building NLP systems are manifold:

- Superior Performance: PTLMs, especially when combined with careful data preprocessing, consistently yield better results than models trained from scratch, as they benefit from extensive exposure to diverse linguistic patterns during pre-training [20].
- Transfer Learning Capability: PTLMs act as a robust foundation, allowing their learned knowledge to be "transferred" to various downstream NLP tasks with minimal fine-tuning, making them highly adaptable to different datasets [20].
- Faster Development and Resource Efficiency: By leveraging pre-trained models, researchers and developers can drastically reduce the time and computational resources required for model training, accelerating the development cycle and making advanced NLP accessible to a wider audience [20].

A wide array of PTLMs are available today, varying in their architectural design (encoder-only, decoder-only, or encoder-decoder) and their pre-training objectives. For abstractive summarization, models based on the encoder-decoder architecture are generally preferred because summarization is inherently a sequence-to-sequence task, transforming an input sequence into a distinct output sequence.

The HuggingFace Transformers library [39] has emerged as a cornerstone in the NLP community, providing an open-source platform with thousands of pre-trained models and datasets. This library simplifies the process of utilizing PTLMs through user-friendly APIs, including "pipelines" that abstract away much of the underlying complexity for common NLP tasks like summarization.

For abstractive summarization, particularly given the criteria of text-to-text conversion and encoder-decoder architecture, three prominent pre-trained models stand out:

1. BART (Bidirectional and Auto-Regressive Transformers):

○ Developed by: Facebook AI (Lewis et al., 2019) [40].

○ Architecture: Encoder-decoder transformer.

○ Pre-training Objective: BART is a "denoising autoencoder." During pre-training, it corrupts text in various ways (e.g., masking spans of text, deleting words, permuting sentences, rotating documents, or filling in tokens randomly). The model is then trained to reconstruct the original, uncorrupted text. This pre-training task forces BART to learn deep contextual representations and develop strong generative capabilities, making it exceptionally effective for tasks that require generating fluent and coherent text, such as summarization, translation, and text completion [40]. Its bidirectional encoder (like BERT) allows it to understand context from both left and right, while its auto-regressive decoder (like GPT) is optimized for text generation.

2. T5 (Text-To-Text Transfer Transformer):

○ Developed by: Google (Raffel et al., 2021) [41].

○ Architecture: Encoder-decoder transformer.

○ Pre-training Objective: T5's core innovation is its "text-to-text" framework, where every NLP problem is reframed as a text-to-text task. For instance, for summarization, the input might be "summarize: [document]" and the output is the summary. This unified approach, combined with pre-training on a massive, cleaned web dataset called "Colossal Clean Crawled Corpus" (C4) using various denoising objectives (similar to BART, but with specific masking strategies), makes T5 highly versatile. It learns to perform diverse NLP tasks by simply generating the appropriate text output. T5 is also notable for being one of the earlier and highly performant Large Language Models (LLMs) [42, 41].

3. PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization):

○ Developed by: Google (Zhang et al., 2020) [43].

○ Architecture: Encoder-decoder transformer.

○ Pre-training Objective: PEGASUS is specifically designed and optimized for abstractive summarization. Its unique pre-training objective is called "Gap Sentence Generation (GSG)." In GSG, whole sentences are removed or "masked" from a document, and the model is trained to generate these missing sentences based on the surrounding context. This task inherently aligns with abstractive summarization, as it teaches the model to generate abstractive content (the "gap sentences") that captures the essence of a larger body of text, using highly salient content. This objective has proven extremely effective in achieving high ROUGE scores for abstractive summarization [43].

When initiating a summarization project, especially one dealing with challenging data like social media, comparing these pre-trained models is a judicious first step, as their specific pre-training objectives might yield varying performance based on the dataset's characteristics. This comparative analysis helps in selecting the most suitable model for fine-tuning.

3.3. Methodology: System Design for Social Media Summarization

Developing an effective abstractive summarization system for social media content requires a carefully designed methodology, acknowledging the unique peculiarities of this data. Social media text is not formal; it is characterized by its brevity, abbreviations, slang, special characters, and emojis. Furthermore, it often contains redundant and repetitive information across multiple posts or comments, which can confuse readers if not handled properly. Therefore, significant emphasis must be placed on data preprocessing and structuring before feeding it into the transformer model.

The proposed methodology for generating summaries of user comments on social media posts, leveraging transformer models, follows a systematic approach comprising several key steps:

3.3.1. Data Collection

The initial and often most challenging step is data acquisition. Publicly available datasets specifically curated for social media summarization are less abundant compared to those for news articles or scientific papers, primarily due to data privacy regulations and platform-specific access limitations.

- Source Selection: The choice of social media platform (e.g., Facebook, Twitter/X, Reddit, Sina Weibo [24]) depends on the research focus and data availability. For this system, a dataset of Facebook news posts accompanied by user comments was utilized.
- Data Structure: Raw social media data typically comes with various fields (e.g., created_time, from_id, from_name, message, post_name, post_title, post_num). Identifying and retaining relevant columns is crucial. In this case, post_title and post_number are important for identifying and grouping specific discussions or events, while the message column contains the actual user comments to be summarized. A large dataset size (e.g., 1,781,576 rows) indicates the scale of information present.

3.3.2. Pre-processing

This is arguably the most critical stage for social media summarization due to the "noisy" nature of the raw data. Pre-processing aims to clean the data from unnecessary elements that could degrade model performance and to normalize it for consistent input. This stage is typically implemented using powerful NLP libraries and regular expressions. Key steps include:

- Punctuation Removal: Standard punctuation (periods, commas, exclamation marks) that might not contribute to core meaning or could interfere with tokenization is removed.
- Special Character Removal: Non-alphanumeric characters, symbols, and artifacts specific to web scraping or social media formatting (e.g., &, <, >) are eliminated.
- Emoji String Handling: Emojis are significant in conveying sentiment and meaning in social media. Depending on the objective, they might be removed, replaced with their textual descriptions (e.g., 😃 -> [HAPPY_FACE]), or handled through specific emoji tokenizers. For summarization, removing them might simplify the task if the model struggles with non-textual elements.
- URL/Hyperlink Removal: URLs, while providing context in original posts, are usually not relevant for the concise textual summary.
- Hashtag and Mention Handling: Hashtags (#topic) and user mentions (@username) are central to social media discourse. They might be removed, or the # and @ symbols could be stripped to treat them as regular words, or they could be kept as special tokens if their semantic value is to be preserved by the model.
- Abbreviation Expansion: While challenging, expanding common abbreviations (e.g., "lol" to "laughing out loud") can improve textual coherence, though it might remove some of the authentic social media flavor.
- Case Normalization: Converting all text to lowercase helps reduce vocabulary size and ensures consistency (e.g., "The" and "the" are treated as the same word).
- NULL Value Handling: Missing or empty entries in data columns must be identified and either removed or imputed.
- Tokenization: Breaking down the cleaned text into individual words or sub-word units (tokens) is essential for numerical representation.
- Stop Word Removal (Optional): Common words like "a," "an," "the" (stop words) often carry little semantic meaning for summarization and can sometimes be removed to reduce noise, though this is less common with modern transformer models that can learn to ignore them.
- Stemming/Lemmatization (Optional): Reducing words to their root form (e.g., "running," "runs," "ran" -> "run") can reduce vocabulary size but might sacrifice some semantic nuance. Modern transformer models are often robust enough not to require this.

3.3.3. Topic-based Data Grouping

Given that social media user comments often revolve around a specific post or event, grouping comments by discussion topic is crucial. This step aggregates related information, providing a richer context for summarization than individual, isolated comments.

- Grouping Mechanism: Based on identifying elements like post_title or post_number, user comments pertaining to the same post are collected into separate lists or discussion threads. This creates distinct "documents" for the summarization model, where each "document" is a collection of comments related to a single topic.
- Dynamic Document Creation: Each grouped

"document" (comment list) can have a variable size, reflecting the varying levels of discussion around different topics.

3.3.4. Feeding the Encoder with Text Lists (Input Representation)

After grouping and preprocessing, the structured data is prepared for the transformer's encoder.

- Tokenization and Numerical Identification: Each word (or sub-word token) in the grouped comment lists (input sequence) is converted into a unique numerical identifier using a pre-trained tokenizer associated with the chosen transformer model (e.g., T5 tokenizer).
- Input IDs and Attention Masks: The numerical tokens form the input_ids. Alongside these, attention_masks are generated. Attention masks are binary tensors (0s and 1s) that tell the model which tokens to attend to (1) and which to ignore (0, typically padding tokens added to make sequences of equal length for batch processing). This ensures that padding does not influence the attention mechanism.
- Embedding Layer: The first layer of the transformer encoder is an embedding layer, which converts these numerical input tokens into dense vector representations. These learned embeddings capture initial semantic meanings.
- Positional Encodings: Positional encodings are added to these embeddings to infuse information about the order of words in the sequence, critical for sequential understanding as attention mechanisms are inherently order-agnostic [5].
- Encoder Stack Processing: The combined embeddings and positional encodings are then passed through the multiple layers of the encoder stack. Each layer processes the input through its Multi-Head Attention and Feed-Forward Network sub-layers, generating a progressively richer and more contextualized representation of the input text. The Multi-Head Attention mechanism, with its multiple attention "heads," allows the model to capture different aspects of word relationships and create various attention matrices, which are then combined to form a comprehensive understanding of the input context.

3.3.5. Passing Data to the Decoder (Output Generation)

The decoder takes the encoder's output (the contextualized representation of the source comments) and generates the summary.

- Decoder Inputs: The decoder typically requires two main inputs:

1. Encoder Output: The final hidden states (or output representation) from the encoder are fed into the decoder's cross-attention mechanism. This allows the decoder to "look back" at the source information while generating the summary.
2. Right-Shifted Output Text (Generated IDs): During training, the decoder is fed the actual target summary (shifted to the right by one token) as input. This "teacher forcing" mechanism allows the decoder to learn the correct next token prediction given the previous correct tokens. During inference (when generating a new summary), the decoder starts with a special "start-of-sequence" token and then feeds its own previously generated tokens as input for the next step.

- Masked Multi-Head Self-Attention: Within the decoder, a masked multi-head self-attention mechanism ensures that when predicting a token, the model only attends to tokens already generated (or the start token), preventing information leakage from future tokens.
- Cross-Attention (Encoder-Decoder Attention): This crucial layer allows the decoder to attend to the output of the encoder. By querying the encoder's keys and values, the decoder can focus on specific parts of the source comments that are most relevant for generating the current word in the summary. This direct interaction between encoder and decoder is key to abstractive generation.
- Output Layer: The final layers of the decoder produce a probability distribution over the entire vocabulary for the next word. A linear layer followed by a softmax function converts the decoder's hidden states into these probabilities.

3.3.6. Generate Summaries (Decoding Strategies)

Once the model is trained, generating summaries involves sampling words based on the decoder's output probabilities. Different decoding strategies influence the quality and diversity of the generated summaries:

- Greedy Search: At each step, the decoder simply picks the word with the highest probability. This is fast but can lead to suboptimal or repetitive summaries.
- Beam Search: At each step, the decoder keeps track of the top-k most probable partial sequences (beams). It then expands these k sequences in the next step, selecting the top k among all possible extensions. This typically yields higher-quality summaries by exploring more possibilities.
- Sampling-based Methods (Top-K, Nucleus Sampling): These methods introduce randomness into the generation process to increase diversity and reduce repetition, often by sampling only from the top K most probable tokens or from a cumulative probability mass P (nucleus sampling).

3.3.7. Summary Validation (Evaluation)

The final step involves evaluating the quality of the generated summaries.

- Quantitative Metrics (ROUGE): The most widely used metric for summarization is ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [45]. ROUGE compares the generated summary to one or more human-written "reference" summaries by counting the overlap of n-grams (sequences of words).

○ ROUGE-1: Measures the overlap of unigrams (single words).

○ ROUGE-2: Measures the overlap of bigrams (two-word sequences).

○ ROUGE-L: Measures the longest common subsequence (LCS) between the generated and reference summaries, which implicitly accounts for sentence-level fluency and order.

○ Precision (P), Recall (R), F1-score (F): For each ROUGE type, precision measures how many words in the generated summary are also in the reference; recall measures how many words in the reference are captured by the generated summary; and F1-score is the harmonic mean of precision and recall. Higher ROUGE scores generally indicate better summary quality, particularly in terms of content overlap [45].

● Qualitative Human Evaluation: While automated metrics like ROUGE are efficient, they don't fully capture aspects like fluency, coherence, factual consistency, or readability. Human evaluation remains the gold standard, where human annotators rate summaries based on criteria such as:

○ Informativeness: Does the summary capture all key points?

○ Fluency: Is the language natural and grammatically correct?

○ Coherence: Do the sentences flow logically and smoothly?

○ Conciseness: Is the summary brief without omitting crucial information?

○ Factuality: Is all information in the summary supported by the source text?

The methodology described provides a robust framework for applying encoder-decoder transformer models to abstractive social media summarization, addressing both the general challenges of summarization and the specific complexities of social media data. The choice of a pre-trained model like T5, BART, or PEGASUS, followed by meticulous preprocessing and evaluation, forms the backbone of a high-performing summarization system.

3.4. Prompt Engineering for Abstractive Summarization

In recent years, especially with the proliferation of increasingly powerful Large Language Models (LLMs) [42], prompt engineering has emerged as a significant technique to guide and optimize the behavior of pre-trained models. This approach involves crafting specific input prompts or instructions to steer the model towards generating desired outputs without the need for extensive fine-tuning of the entire model parameters [27, 32]. While traditional fine-tuning on task-specific datasets remains a primary method for summarization, prompt-based approaches offer a flexible and often more resource-efficient alternative, particularly for adapting to new summarization styles or domains.

The essence of prompt engineering for abstractive summarization lies in designing an input that clearly signals to the LLM what kind of summary is expected. This could involve:

● Instruction-based Prompts: Directly instructing the model to summarize. For example, "Summarize the following social media discussion in under 100 words, focusing on key opinions: [discussion text]".

● Few-Shot Learning (In-Context Learning): Providing a few examples of input text and desired summaries within the prompt itself. This allows the model to infer the task and desired output format from the examples, without updating its weights.

● Prefix-Tuning: A parameter-efficient alternative to full fine-tuning [29]. Instead of updating all model parameters, prefix-tuning involves adding a small, task-specific continuous vector (the "prefix") to the input of each transformer layer. Only this prefix is optimized for a new task, while the vast majority of the pre-trained model's parameters remain frozen. This makes models much more portable and adaptable across many tasks without requiring a full copy of the model for each.

● Prompt-Tuning: Similar to prefix-tuning but even more lightweight [30]. It involves learning a small, continuous "soft prompt" that is prepended to the input embeddings, without modifying the underlying model. This learned prompt guides the frozen LLM to perform the desired task. For very large models, prompt-tuning has shown comparable performance to full fine-tuning, significantly reducing storage and deployment costs.

● Learned Entity Prompts: For abstractive summarization, specifically, techniques like "Planning with Learned Entity Prompts" have been introduced [28]. This involves training a model to generate an "entity chain" (a sequence of important entities and relations) as an intermediate step, which then serves as an additional prompt to guide the final summary generation. This helps ensure that key entities are covered and the summary is grounded in factual elements.

● Structured Prompts with Delimiters: Using specific delimiters or tags within the prompt to separate different parts of the input (e.g., <document>...</document>, <summary_instruction>...). This provides clear boundaries for the model to understand the input structure.

Relevance to Social Media Summarization:

Prompt engineering holds particular promise for social media summarization due to its informal and diverse nature:

1. Style and Tone Control: Prompts can be designed to elicit summaries that match a specific tone (e.g., "Summarize humorously," "Provide a neutral summary") or style, which is crucial for nuanced social media content.

2. Focus Control: Users could specify what aspects of a social media discussion they want to focus on (e.g., "Summarize opinions on climate change," "Highlight criticisms of the new policy"), guiding the model to extract specific information without needing to re-train.

3. Handling Specificities: Prompts can implicitly or explicitly guide the model to better handle social media specificities like hashtags or emojis, by providing examples in the prompt or by instructing the model on how to interpret them.

4. Personalization: With techniques like prefix-tuning or prompt-tuning, it becomes feasible to create personalized summarizers for individual users. A separate, small prefix could be trained for each user based on their specific summarization preferences (e.g., preferred length, level of detail, focus areas), allowing the production of highly specialized and user-tailored summaries [29].

5. Efficiency with LLMs: For incredibly large models (LLMs) where fine-tuning is computationally prohibitive, prompt engineering (especially prompt-tuning and prefix-tuning) offers an efficient way to adapt these powerful general-purpose models to specific summarization tasks without incurring massive retraining costs. This is particularly relevant in fields like healthcare, where traditional NLP methods struggled, and prompt engineering with LLMs has shown significant breakthroughs [31].

In essence, prompt-based learning represents a new and promising frontier in NLP. It allows for a flexible and powerful way to interact with and steer pre-trained language models, bridging the gap between the complexity of existing deep learning techniques and the nuanced requirements of natural language understanding and generation, making it highly applicable to the intricate task of social media summarization [32].

4. Results and Validation

The implementation of the abstractive summarization system, utilizing the robust encoder-decoder transformer architecture, was conducted using a dataset of user posts and comments collected from Facebook. The computational environment for this development comprised a machine equipped with an NVDIA GeForce 4070 GPU, featuring 12 GB of RAM. This hardware configuration is a common choice for deep learning tasks, offering a balance of computational power and memory for handling moderately sized models and datasets.

Given the substantial volume of the raw dataset, which originally contained 1,781,576 rows, and considering the available computational resources, a strategic decision was made to utilize a pre-processed subset of this data. Specifically, approximately one-third of the obtained dataset was employed for the training and validation phases of the model. This sampling approach is a practical measure to manage computational load while still providing sufficient data for the model to learn effectively.

The methodology emphasized the grouping of data by discussion topic, an essential step to ensure that summaries are generated from cohesive conversational threads rather than isolated comments. This topic-based grouping allowed the creation of distinct "documents" for the model, each representing a complete set of user comments related to a particular Facebook post.

For training and evaluation, the processed dataset was meticulously divided into three subsets:

- Training Set: 80% of the data, used to train the model and adjust its internal parameters.
- Validation Set: 20% of the data, used during training to monitor the model's performance on unseen data and to tune hyperparameters (e.g., learning rate, dropout). This set helps prevent overfitting.
- Test Set: (Validation/2) = 10% of the original data (meaning half of the validation set was designated as the test set in this specific setup). This set is held out completely during training and validation and is used only once at the very end to provide an unbiased evaluation of the final model's generalization capability.

The model selected for this implementation was the pre-trained T5-base model. T5-base is a variant of the T5 architecture, which is known for its "text-to-text" paradigm, unifying all NLP tasks into a single text-based format. The specific configuration of the T5-base model used included:

- 12 layers: Referring to the Nx stack of encoder and decoder layers within the transformer architecture.
- 12 attention heads: Indicating the number of parallel attention mechanisms in each multi-head attention layer, allowing the model to capture diverse relationships within the data.
- Network feed depth of 3072: This refers to the dimensionality of the hidden layer in the position-wise feed-forward networks within each transformer layer.

The training process was configured with the following hyperparameters:

- Optimizer: AdamW, a widely used optimization algorithm for deep learning models, known for its effectiveness with transformers.
- Learning Rate: 1e−3 (0.001), a common starting point for learning rates in transformer training.
- Dropout Rate: 0.1, a regularization technique applied to prevent overfitting by randomly setting a

fraction of input units to zero at each update during training.

- Epochs: The model was trained for 12 epochs, meaning the entire training dataset was passed through the neural network 12 times.
- Batch Size: 64, indicating that 64 samples were processed at a time before the model's weights were updated.

4.1. Loss Analysis

A crucial aspect of evaluating the model's performance during training is monitoring the loss function. Loss measures the discrepancy between the model's predictions and the actual target values; a lower loss indicates better model performance. Both training loss and validation loss were tracked across the 12 epochs.

The observation of a gradual decrease in both training and validation loss is a highly positive indicator. This suggests that:

- Model Learning: The model is effectively learning from the data, progressively reducing its prediction errors over successive epochs.
- Generalization: The validation loss also decreasing and the gap between training and validation loss shrinking after each epoch is particularly important. A small and converging gap indicates that the model is not merely memorizing the training data (overfitting) but is learning generalizable patterns that apply well to unseen data. This suggests that the model is shrinking its regularization loss (related to model weights), leading to a small difference between the validation and train loss, which is a hallmark of good generalization. If the validation loss started to increase while training loss continued to decrease, it would signal overfitting, necessitating adjustments to training.

The consistent reduction and convergence of both loss metrics provide confidence in the training process and the model's ability to learn effectively from the social media comment dataset.

4.2. ROUGE Score Results

To quantitatively assess the quality of the generated summaries, the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) metric was employed [45]. ROUGE is the industry standard for evaluating automatic summarization systems. It works by measuring the n-gram overlap between the system-generated summary and a set of human-written "reference" summaries. In this specific research, the ROUGE comparison was made between the system-generated summary and each comment in the thread serving as fragmented references.

Metric	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Precision (P)	0.520	0.514	0.519
Recall (R)	0.854	0.849	0.854
F1-score (F)	0.646	0.640	0.645

Interpretation of ROUGE Scores:

- Precision: Represents the ratio of overlapping n-grams in the generated summary to the total number of n-grams in the generated summary. For instance, a ROUGE-1 Precision of 0.520 means that 52% of the unigrams in the generated summary are also present in the reference comments.
- Recall: Represents the ratio of overlapping n-grams in the generated summary to the total number of n-grams in the reference comments. A ROUGE-1 Recall of 0.854 indicates that the generated summary captured 85.4% of the unigrams present in the reference comments. High recall suggests that the generated summary covers most of the information in the source.
- F1-Score: The harmonic mean of precision and recall. It provides a balanced measure that considers both how much relevant information is included (recall) and how much irrelevant information is excluded (precision). An F1-score of around 0.64 indicates a reasonably good balance between precision and recall for the generated summaries.

Analysis of the Specific Scores:

- The high Recall scores (0.849-0.854) across ROUGE-1, ROUGE-2, and ROUGE-L suggest that the T5 model is highly effective at capturing a significant portion of the information present in the source comments. This is a crucial aspect for summarization, ensuring that the generated summary is informative.
- The lower Precision scores (0.514-0.520) compared to recall might indicate that while the model captures a lot of relevant information, it might also generate some content not strictly present in the reference comments or that is less concise than ideal. However, in abstractive summarization, perfectly matching reference n-grams is not always the goal, as the model generates novel sentences.

- The F1-scores (0.640-0.645) represent a good overall performance. ROUGE scores typically range from 0 to 1, with 1 being a perfect match. Achieving F1-scores in the 0.60-0.65 range for abstractive summarization of informal social media content is generally considered satisfactory, particularly given the inherent difficulty of the task and the fragmented nature of the reference (individual comments).

It is noted that the T5 model demonstrated the highest performance among the three compared pre-trained models (BART, T5, and PEGASUS) on the specific dataset used for this work. This finding aligns with the general understanding of T5's strong capabilities in various text-to-text generation tasks, including summarization, and its position as an early, high-performing LLM [42]. The results indicate that the transformer-based encoder-decoder architecture, particularly with a well-chosen and fine-tuned pre-trained model like T5, is a viable and effective solution for abstractive summarization of social media content.

5. Discussion: Challenges and Future Directions

The application of encoder-decoder transformer models to abstractive social media summarization has yielded promising results, demonstrating their capability to process complex, noisy, and informal textual data to generate coherent and informative summaries. The high recall scores observed in the ROUGE evaluation suggest that the model effectively captures a significant portion of the key information from social media comment threads. However, as with any rapidly evolving research area, significant challenges persist, and numerous opportunities for future advancements remain.

5.1. Challenges

Despite the strengths of transformer models, their deployment for social media summarization faces several inherent difficulties that warrant deeper discussion:

5.1.1. Handling Extreme Brevity and Noise

Social media content is characterized by its extreme brevity, informal language, abbreviations, misspellings, and platform-specific jargon. While pre-trained transformers are robust, they still struggle with the nuances of highly fragmented, grammatically unconventional, and lexically sparse input. The challenge lies not just in understanding individual noisy posts, but in synthesizing information from a collection of such posts that may individually lack sufficient context but collectively form a meaningful discussion.

- Ambiguity and Context: Slang, memes, and context-dependent expressions can be highly ambiguous outside of their immediate social media context. Models may misinterpret these or fail to capture their full semantic weight, leading to less accurate or less nuanced summaries [14, 15, 20].
- Informal Norms: Social media often bypasses formal grammatical rules. While models can learn from large datasets, generating summaries that are both grammatically correct (for readability) and accurately reflect the informal tone or intent of the original posts is a fine balance.
- Lexical Scarcity: Individual posts might have limited vocabulary, making it difficult for models to identify key terms or concepts that are only implicitly conveyed across multiple short interactions.

5.1.2. Real-time and Dynamic Content Processing

Social media streams are continuously updated, especially during breaking news or trending events. Summarizing these rapidly evolving feeds in real-time presents considerable computational and algorithmic challenges.

- Latency: Generating abstractive summaries is computationally intensive. Ensuring low latency for real-time applications, where summaries need to be updated almost instantaneously as new content emerges, requires highly optimized models and infrastructure.
- Incremental Summarization: Most transformer models are designed to process a fixed input and generate a static summary. Developing robust methods for incremental summarization, where the summary is dynamically updated as new posts arrive without re-processing the entire historical data, is a significant challenge [26]. This requires models that can efficiently incorporate new information while maintaining the coherence and consistency of previous summaries.
- Event Detection and Tracking: Before summarization, accurately detecting and tracking the evolution of an event within a social media stream is crucial for providing timely and relevant summaries [11, 12, 23]. Models need to understand when an event starts, changes, or concludes to provide appropriate summaries.

5.1.3. Multimodal Summarization

Modern social media platforms are inherently multimodal, integrating text with images, videos, audio, and links. A comprehensive summary of a social media event or discussion might necessitate drawing information from all these modalities.

- Cross-Modal Understanding: Current transformer models excel at text-to-text tasks. Integrating information from visual (images, videos) or audio modalities requires sophisticated multimodal learning architectures that can process and fuse information from disparate data types.
- Coherent Multimodal Output: Generating a text summary that accurately reflects insights derived from non-textual content, or generating a multimodal summary (e.g., text with key images), is an active and complex research area [8, 9, 10]. For example, summarizing a live event might require extracting key textual descriptions, identifying relevant images, and even summarizing audio from attached videos.

5.1.4. Factuality and Hallucination

Abstractive models, by their nature, generate novel text, which can sometimes lead to "hallucinations"—the generation of plausible-sounding but factually incorrect or unsupported information not present in the source text [1]. This is a critical concern, particularly when summarizing sensitive topics, news, or user-generated factual claims on social media.

- Verifiability: Unlike extractive summaries, where information can be directly traced back to the source, abstractive summaries lack easy verifiability, making it harder to spot and correct inaccuracies.
- Misinformation Spread: In the context of social media, where misinformation can spread rapidly, a hallucinating summarizer could inadvertently amplify or create false narratives, posing significant ethical and societal risks.
- Trust and Reliability: For a summarization system to be widely adopted, especially in critical applications (e.g., public safety, crisis management), its output must be highly trustworthy and factually grounded.

5.1.5. Bias and Ethical Considerations

Large pre-trained language models are trained on vast datasets that often reflect societal biases (e.g., gender, racial, political biases) present in the internet text from which they are sourced. These biases can be inadvertently reflected or even amplified in the generated summaries.

- Representational Bias: Summaries might unfairly represent certain viewpoints, marginalize minority opinions, or stereotype groups if the training data was skewed.
- Harmful Content: Models might inadvertently generate or retain harmful content (e.g., hate speech, discriminatory language) if it was present in the source and the model was not specifically trained to filter or neutralize it.
- Transparency and Accountability: The "black box" nature of deep learning models makes it challenging to understand why a particular summary was generated, raising questions about accountability when errors or biases occur.
- Privacy Concerns: Summarizing user-generated content raises privacy concerns, especially if personally identifiable information is inadvertently included in summaries or if the summarization process itself involves processing sensitive data.

5.2. Future Directions

Addressing the aforementioned challenges and pushing the boundaries of social media summarization offers numerous exciting avenues for future research:

5.2.1. Enhanced Domain Adaptation and Specialized Models

While general pre-trained models are powerful, fine-tuning them on highly specific social media datasets could significantly enhance their performance and relevance for particular domains (e.g., summarizing discussions in medical forums, political debates, or customer reviews) [20, 24].

- Parameter-Efficient Fine-Tuning (PEFT): Exploring more advanced PEFT techniques like LoRA (Low-Rank Adaptation) alongside prompt-tuning [30] or adapter modules could allow for efficient adaptation to new social media sub-domains or stylistic requirements without the need to retrain large portions of the model.
- Continual Learning: Developing models that can continuously learn and adapt from new social media data streams without forgetting previously acquired knowledge (catastrophic forgetting) is crucial for dynamic environments.

5.2.2. Long-Document Summarization for Social Media Aggregations

Social media data related to an event can accumulate into extremely long "documents" (e.g., thousands of comments). Most current transformer models, including BART and T5, have limitations on input sequence length due to the quadratic complexity of standard self-attention with respect to sequence length.

- Efficient Attention Mechanisms: Research into more efficient attention mechanisms (e.g., sparse attention, linear attention, BigBird, Longformer) that scale linearly or logarithmically with sequence length is critical for processing very long social media aggregations.
- Hierarchical Summarization: Developing multi-stage or hierarchical summarization approaches, where an initial stage condenses local chunks of text (e.g., individual comment threads), and a subsequent stage summarizes these condensed representations into a global summary, could be highly effective [35]. This mirrors how humans might summarize a very long document.
- Pre-training for Long Sequences: Developing new pre-training objectives specifically designed for long-document summarization could yield models better equipped to handle extensive social media archives.

5.2.3. Integration with Large Language Models (LLMs)

The rapid advancements and unprecedented generative capabilities of ultra-large LLMs (e.g., GPT-3, GPT-4, Gemini) present new frontiers for social media summarization [42].

- Advanced Prompting Techniques: Leveraging sophisticated prompt engineering strategies [31, 32] (e.g., chained prompts, self-consistency prompting, tree-of-thought prompting) could enable LLMs to produce highly nuanced and contextually aware social media summaries without requiring extensive fine-tuning.

- Multi-Stage Architectures with LLMs: LLMs could serve as powerful "backend" abstractive engines, taking input from an initial stage that extracts salient information from raw social media streams, possibly using smaller, more efficient models.

- Knowledge Grounding: Integrating LLMs with external knowledge bases to ensure factual accuracy and reduce hallucination by grounding the generated summaries in verified information.

5.2.4. User-Guided and Interactive Summarization

To enhance the practical utility and user satisfaction of social media summarization systems, future research should focus on making them more interactive and customizable.

- User Preferences: Allowing users to specify desired summary length, level of detail, desired focus (e.g., sentiment, key entities, specific arguments), or target audience can tailor summaries to individual needs.

- Iterative Refinement: Developing interfaces where users can provide feedback on a generated summary (e.g., "make this section longer," "remove this topic") and the model iteratively refines its output.

- Personalized Summarization: Building user profiles based on their consumption habits and summarization preferences, and then training (or adapting with prompt-tuning) models to generate summaries that inherently align with those preferences [29].

5.2.5. Beyond ROUGE: Comprehensive Evaluation Metrics

While ROUGE [45] is widely used for its simplicity and automation, it primarily measures n-gram overlap and doesn't fully capture crucial aspects of abstractive summary quality such as factual consistency, overall coherence, grammatical correctness, or readability.

- Factual Consistency Metrics: Developing automated metrics that can reliably assess whether the generated summary contains information that is factually consistent with the source document.

- Coherence and Fluency Metrics: Research into better automated measures of linguistic quality that go beyond n-gram overlap, possibly leveraging other language models.

- Human-Centric Evaluation: Continued emphasis on robust human evaluation protocols, possibly involving crowdsourcing, to get more reliable assessments of subjective quality aspects. Comparing system-generated summaries against human-written summaries (rather than individual comments as fragmented references) is critical for a more accurate assessment of abstractive quality, as highlighted in the provided PDF's discussion.

- Task-Specific Evaluation: Evaluating summaries based on their utility for specific downstream tasks (e.g., how well they enable users to answer questions about the discussion, or how quickly they convey key information).

The journey to perfect abstractive social media summarization is ongoing. By addressing these challenges and exploring these promising directions, future research can unlock the full potential of transformer models to transform how we consume and understand the vast and dynamic landscape of social media information.

6. CONCLUSION

The exponential growth of information on social media platforms, driven by widespread smart technology adoption, has made it practically impossible for individuals to manually process the daily deluge of posts and comments. In this digital era, social media has become the dominant conduit for information, opinion exchange, and the expression of diverse social, psychological, economic, and political beliefs. The sheer volume of user-generated content, especially comments often proportional to a post's importance, necessitates effective summarization to provide accurate, timely, and digestible insights. User comments, in particular, are invaluable as they reflect public opinion, a critical understanding for both content creators and consumers.

However, summarizing social media content is a complex undertaking due to its inherent specificities: it is often informal, exhibits linguistic deficiencies, lacks lexical richness within individual entries, and is replete with abbreviations, slang, special symbols, and emojis. Traditional text summarization methods, designed for formal, well-structured documents, often fall short when confronted with these peculiarities.

Recent advancements in Natural Language Processing, particularly the advent of transformer models, have revolutionized the landscape of text summarization. These models, with their powerful attention mechanisms, have demonstrated a remarkable aptitude for processing natural language, yielding excellent results in generating coherent and human-like summaries. In the context of social media, transformer-based approaches have emerged as the most promising technology to effectively manage the challenging nature of its content.

This article has comprehensively explored the application of encoder-decoder transformer models for abstractive summarization of social media user comments. We delved into the fundamental architecture of transformers, highlighting how their multi-head attention and parallel processing capabilities enable them to capture long-range dependencies and generate highly contextualized representations of input text. We discussed prominent pre-trained encoder-decoder models—BART, T5, and PEGASUS—each with unique pre-training objectives tailored for generative tasks. Our analysis, consistent with findings in the broader research community, indicated that the T5 model demonstrated superior performance on the utilized social media comment dataset, evidenced by robust ROUGE metrics. T5's effectiveness is further bolstered by its classification as an early, high-performing Large Language Model, showcasing its proficiency in language generation.

The evaluation through loss curves demonstrated a sound learning process, with training and validation losses converging, indicating good generalization capability. The ROUGE scores, particularly the high recall, confirmed the model's ability to capture a significant amount of relevant information from the informal and often fragmented social media discussions.

Despite these successes, the field of social media summarization with transformers is ripe for further development. Key challenges include consistently ensuring factual consistency and mitigating "hallucinations," effectively integrating multimodal content (images, videos) into summaries, optimizing for real-time and dynamic data streams, and addressing potential biases inherent in large language models. Future research efforts should concentrate on refining models for greater robustness to social media noise, developing efficient methods for long-document summarization (e.g., extensive comment threads), harnessing the full potential of even larger language models through advanced prompt engineering, and creating user-guided interactive summarization systems. Furthermore, the development of more comprehensive evaluation metrics beyond simple n-gram overlap is crucial to truly assess the quality of abstractive, human-like summaries.

As social media continues to be an indispensable source of real-time information and public discourse, the role of advanced abstractive summarization, powered by sophisticated encoder-decoder transformer models, will only grow in importance. These technologies are crucial for enabling individuals and organizations to navigate, comprehend, and extract valuable insights from the ever-expanding digital landscape more effectively and efficiently.

REFERENCES

[1] Gupta, S. and Gupta, S. K. Abstractive summarization: An overview of the state of the art. Expert Systems with Applications 121, 2019, pp. 49–65.https://doi.org/10.1016/j.eswa.2018.12.011

[2] Luhn, H. P. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, vol. 2, no. 2, Apr. 1958, pp. 159-165,https://doi: 10.1147/rd.22.0159

[3] Suleiman, D., A. Awajan, A. Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges. Mathematical Problems in Engineering, 2020,https://doi.org/10.1155/2020/9365340

[4] Gupta, V., Lehal, G. S. A Survey of Text Summarization Extractive techniques. Journal of Emerging Technologies in Web Intelligence, 2010, pp. 258–268,https://doi.org/10.4304/jetwi.2.3.258-268

[5] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. Attention Is All You Need In 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA., June 2017.https://arxiv.org/abs/1706.03762

[6] Sharifi, B., Hutton, M-A. and Kalita, J. (2010) “Summarizing Microblogs Automatically”. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 685–688, Los Angeles, California. Association for Computational Linguistics.

[7] Sharifi, B., Inouye, D., and Kalita, J.K. (2014) “Summarization of Twitter Microblogs”. The Computer Journal, Volume 57, Issue 3, March 2014, Pages 378–402,https://doi.org/10.1093/comjnl/bxt109

[8] F. Amato, F. Moscato, V. Moscato, A. Picariello, G. Sperli’, “Summarizing social media content for multimedia stories creation”. The 27th Italian Symposium on Advanced Database Systems (SEB 2019).https://ceur-ws.org/Vol-2400/paper-40.pdf

[9] F. Amato, A. Castiglione, F. Mercorio, M. Mezzanzanica, V. Moscato, A. Picariello, G. Sperlì, “Multimedia story creation on social networks,” Future Generation Computer Systems, 86, 412–420, 2018,https://doi.org/10.1016/j.future.2018.04.006

[10] J. Bian, Y. Yang, H. Zhang, T. S. Chua, “Multimedia summarization for social events in microblog stream,” IEEE Transactions on Multimedia, 17(2), 216–228, 2015,https://doi.org/10.1109/TMM.2014.2384912

[11] D. Chakrabarti, K. Punera, “Event Summarization Using Tweets”. Proceedings of the International AAAI Conference on Web and Social Media, 5(1), 66-73.https://doi.org/10.1609/icwsm.v5i1.14138. 2011

[12] Chong, F., Chua, T., Asur, S. (2021) “Automatic Summarization of Events from Social Media”. Proceedings of the International AAAI Conference on Web and Social Media, 7(1), 81-90.https://doi.org/10.1609/icwsm.v7i1.14394

[13] Gao, S., Chen, X., Li, P., Ren, Z., Bing, L., Zhao, D. and Yan, R. (2019) “Abstractive Text Summarization by Incorporating Reader Comments”. In The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), 33(01): 6399-6406.https://doi.org/10.1609/aaai.v33i01.33016399

[14] Liang, Z., Du, J. and Li, C. (2020) “Abstractive social media text summarization using selective reinforced Seq2Seq attention model,” Neurocomputing, 410, 432–440,https://doi.org/10.1016/j.neucom.2020.04.137

[15] Wang, Q. and Ren, J. (2021) “Summary-aware attention for social media short text abstractive summarization,” Neurocomputing, 425, 290–

299,https://doi.org/10.1016/j.neucom.2020.04.136

[16] Bhandarkar, P., Thomas, K. T. (2023) "Text Summarization Using Combination of Sequence-To-Sequence Model with Attention Approach", Springer Science and Business Media Deutschland GmbH: 283–293, 2023,https://doi.org/10.1007/978-981-19-3035-5_22

[17] Gupta, A., Chugh, D. and Katarya, R. (2022) "Automated News Summarization Using Transformers", In Sustainable Advanced Computing, 2022, Volume 840. ISBN: 978-981-16-9011-2.https://arxiv.org/pdf/2108.01064

[18] M. H. Su, C. H. Wu, H. T. Cheng, "A Two-Stage Transformer-Based Approach for Variable-Length Abstractive Summarization," IEEE/ACM Transactions on Audio Speech and Language Processing, 28, 2061–2072, 2020,https://doi.org/10.1109/TASLP.2020.3006731

[19] D. Singhal, K. Khatter, A. Tejaswini, R. Jayashree, "Abstractive Summarization of Meeting Conversations," in 2020 IEEE International Conference for Innovation in Technology, INOCON 2020, Institute of Electrical and Electronics Engineers Inc., 2020,https://doi.org/10.1109/INOCON50539.2020.9298305

[20] Ivan S. Blekanov, Nikita Tarasov and Svetlana S. Bodrunova. 2022. Transformer-Based Abstractive Summarization for Reddit and Twitter: Single Posts vs. Comment Pools in Three Languages. Future Internet 14, 69.https://doi.org/10.3390/fi14030069

[21] A. Pal, L. Fan, V. Igodifo, Text Summarization using BERT and T5.https://anjali001.github.io/Project_Report.pdf

[22] M. T. Nguyen, V. C. Nguyen, H. T. Vu, V. H. Nguyen, "Transformer-based Summarization by Exploiting Social Information," in Proceedings - 2020 12th International Conference on Knowledge and Systems Engineering, KSE 2020, Institute of Electrical and Electronics Engineers Inc.: 25–30, 2020,https://doi.org/10.1109/KSE50997.2020.9287388

[23] Li, Q. and Zhang, Q. (2020) "Abstractive Event Summarization on Twitter". In The Web Conference 2020 - Companion of the World Wide Web Conference, WWW 2020, Association for Computing Machinery: 22–23,https://doi.org/10.1145/3366424.3382678

[24] Z. Kerui, H. Haichao, L. Yuxia, "Automatic text summarization on social media," in ACM International Conference Proceeding Series, Association for Computing Machinery, 2020,https://doi.org/10.1145/3440084.3441182

[25] Tampe, I., Mendoza, M. and Milios, E. (2021) "Neural Abstractive Unsupervised Summarization of Online News Discussions". In: Arai, K. (Eds) Intelligent Systems and Applications. IntelliSys 2021. Lecture Notes in Networks and Systems, vol 295. Springer, Cham.https://arxiv.org/abs/2106.03953

[26] Rawat, R., Rawat, P., Elahi V. and Elahi, A. (2021) "Abstractive Summarization on Dynamically Changing Text," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1158-1163,https://doi.org/10.1109/ICCMC51019.2021.9418438

[27] Ding N, Hu S, Zhao W, Chen Y, Liu Z, Zheng H, et al. OpenPrompt: An Open-source Framework for Prompt-learning. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations; 2022. p. 105-13.https://doi.org/10.18653/v1/2022.acl-demo.10

[28] Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simões, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with Learned Entity Prompts for Abstractive Summarization. Transactions of the Association for Computational Linguistics, 9: 1475–1492.https://doi.org/10.1162/tacl_a_00438

[29] Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.https://doi.org/10.18653/v1/2021.acl-long.353

[30] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.https://doi.org/10.18653/v1/2021.emnlp-main.243

[31] Wang, Jiaqi, Enze Shi, Sigang Yu, Zihao Wu, Chong Ma, Haixing Dai, Qiushi Yang, Yanqing Kang, Jinru Wu, Huawen Hu, Chenxi Yue, Haiyang Zhang, Yi-Hsueh Liu, Xiang Li, Bao Ge, Dajiang Zhu, Yixuan Yuan, Dinggang Shen, Tianming Liu and Shu Zhang. "Prompt Engineering for Healthcare: Methodologies and Applications."https://arxiv.org/html/2304.14670v2.

[32] Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi and Graham Neubig. "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing." ACM Computing Surveys 55(2021): 1-35.https://doi.org/10.1145/3560815

[33] Chua, F., Asur, S. Automatic Summarization of Events from Social Media. In Proceedings of the International AAAI Conference on Web and Social Media, 2023, 7(1), pp. 81-90.https://ojs.aaai.org/index.php/ICWSM/article/view/

14394

[34] El-Kassas, W. S., Salama, C. R., Rafea, A. A. and Monhamed, H. K. Automatic Text Summarization: A Comprehensive Survey. Expert Systems with Applications. (2020). 165, 113679.https://doi.org/10.1016/j.eswa.2020.113679

[35] Wang, S., Zhao, X., Li, B., Ge, B. & Tang, D. Integrating extractive and abstractive models for long text summarization. IEEE International Congress on Big Data (Big Data Congress), 2017, pp. 305-312.https://ieeexplore.ieee.org/document/8029339

[36] Varma, V., Kurisinkel, J., Radhakrishnan, P. Social Media Summarization, Cambria, E., Das, D., Bandyopadhyay, S., Feraco, A. (eds) A Practical Guide to Sentiment Analysis. Socio-Affective Computing, vol 5. Springer, Cham., 2017, pp 135–153https://doi.org/10.1007/978-3-319-55394-8_7

[37] Lin, H., & Ng, V. Abstractive Summarization: A Survey of the State of the Art. In Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(01), pp. 9815-9822.https://doi.org/10.1609/aaai.v33i01.33019815

[38] K. Pipalia, R. Bhadja, M. Shukla, “Comparative analysis of different transformer based architectures used in sentiment analysis,” in Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020, Institute of Electrical and Electronics Engineers Inc.: 411–415, 2020,https://doi.org/10.1109/SMART50582.2020.9337081

[39] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, “HuggingFace’s Transformers: State-of-the-art Natural Language Processing,” 2019.https://doi.org10.18653/v1/2020.emnlp-demos.6

[40] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L. (2019) “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, pages 7871–7880, Online. Association for Computational Linguistics.https://aclanthology.org/2020.acl-main.703

[41] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narag, S., Matena, M., Zhou, Y., Li, W. and Liu P. J. (2021) “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In The Journal of Machine Learning Research, Volume 21, Issue 1, 2019. ISSN: 1532-4435.https://dl.acm.org/doi/abs/10.5555/3455716.3455856

[42] Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Barnes, N., & Mian, A. S. (2023). A Comprehensive Overview of Large Language Models.http://arxiv.org/pdf/2307.06435

[43] Zhang, J., Zhao, Y., Saleh, M. and Liu, P. J. (2020) “PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization”. In ICML'20: Proceedings of the 37th International Conference on Machine Learning, July 2020, Article No.: 1051, Pages 11328–11339.https://dl.acm.org/doi/abs/10.5555/3524938.3525989

[44] Rawat A., and Singh Samant, S. (2022) "Comparative Analysis of Transformer based Models for Question Answering". 2nd International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, India, 2022, pp. 1-6,https://doi.org/10.1109/CISCT55310.2022.10046525

[45] Lin, C.-Y. (2004) “ROUGE: A Package for Automatic Evaluation of Summaries”. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.https://arxiv.org/abs/1803.01937