

Social Media Text Summarization: A Survey Towards a Transformer-based System Design

Afrodite Papagiannopoulou*, Chrissanthi Angeli

School of Engineering, University of West Attica, Athens, 122 43, Greece

ARTICLE INFO

Article history:

Received: 19 September, 2023

Accepted: 13 November, 2023

Online: 30 November, 2023

Keywords:

Social Media Summarization

Natural Language Generation

Neural Networks

Transformers

ABSTRACT

Daily life is characterized by a great explosion of abundance of information available on the internet and social media. Smart technology has radically changed our lives, giving a leading role to social media for communication, advertising, information and exchange of opinions. Managing this huge amount of data by humans is an almost impossible task. Adequacy of summarizing texts is therefore urgently needed, in order to offer people knowledge and information avoiding time-consuming procedures. Various text summarization techniques are already widely used. Artificial intelligence techniques for automated text summarization are a major undertaking. Due to the recent development of neural networks and deep learning models like Transformers, we can create more efficient summaries. This paper reviews text summarisation approaches on social media and introduces our approach towards a summarization system using transformers.

1. Introduction

Today's abundance of information on the Internet and social media makes it imperative to create summaries to keep readers informed in an accurate and timely manner. Social media posts are based on events. When an event breaks out, a huge amount of posts flood social media platforms. These posts can be documents, articles, discussions and conversations on different topics and events, that are time-consuming to read. Therefore, text summarization is essential for retrieving useful knowledge in a reasonable amount of time.

We can proceed summarization in two ways: a) writing a summary manually, which is time-consuming, b) writing algorithms and performing artificial intelligence techniques, which requires much less time. This method is called Automatic Summarization, whose aim is to create a shorter text of the original document, without losing its meaning intact [1,2]

Many algorithms have been applied to generate text summaries. We can consider two main classes of algorithms: Pre-neural, that do not make use neural networks and b) Deep Learning or Machine Learning techniques that make use of neural networks. The later have been successfully applied to various NLP tasks, yielded excellent results, and have been extensively used in recent years [3]. Various deep learning models have been

used, mainly based on both recurrent neural networks (RNN) and convolutional neural networks (CNN). These models have proven highly satisfactory in predicting complex relationships that simple structured or semantic approaches cannot do alone [4]. In recent years, the use of Transfer learning and Transformer models has gained popularity mainly in the field of Natural Language Understanding, Processing and Generation [5].

1.1 Text Summarization Methods

The aim of text summarization is to convert a large text document into a shorter one by preserving the critical information and ensuring the meaning of the text. Due to large amounts of data available on social media, it is almost impossible for anyone to read all the comments generated by the users under a post. It is therefore necessary to properly code and program machines so that they can create coherent summaries just like humans. The process of text summarization done by machines or artificial intelligence programs is known as "Automatic Text Summarization". Automatic text summarization is the process of generating a text summary which, using artificial intelligence techniques and deep learning algorithms, is based on the original text while preserving the information content and overall meaning [1, 6]. Automatic text summarization presents some challenges: First, appropriate information must be selected from the original document so that the summary preserves the meaning of the document. Second, the output text must be fluid and coherent and also expressed in a way that is direct and understandable to the

*Corresponding Author: Afrodite .Papagiannopoulou, apapagiannop@uniwa.gr

reader [2]. We recognize two main categories of text summarization:

- *Extractive summarization* extracts important words and phrases from the original text, uses them as they are and, gathering them together, with a slight rearrangement, generates the summary. [1].
- *Abstractive summarization* abstracts the meaning of the original text and, using paraphrasing, creates a new shorter text which looks completely different but essentially has the same meaning as the original text [1].

1.2 Importance of Social Media Summarization.

The role of social media is becoming increasingly important, especially in public life, as posts and comments made by users are important in shaping public opinion on various issues. These topics refer to economics, politics, entertainment, education, psychology and society. Social media data excels at extracting sentiment and patterns of social behaviour that can be used for social research, economic and political decision-making. So creating summaries seems vital. There is a lot of information flooding the internet and most of it is redundant or repetitive resulting in confusion for readers. For this reason, a mechanism is needed to select the necessary information to provide correct, fast and accurate information.

Recent years have shown that abstractive summarization has achieved great results in the field of document summarization by producing more human-like summaries. Unlike formal documents, online conversations and web communication present three great challenges: 1) tend to be informal, consisting of slang expressions and special characters, 2) show deviations from the original theme and dependencies on previous opinions and, 3) since they are short, they lack lexical richness.

It is important to emphasize that, in the field of social media summarization, few research works [7, 8, 9] have demonstrated transformer-based social media abstractive text summarization systems. Our system aims to optimize the results achieved so far in two main points. First, we focus on creating coherent and meaningful summaries of user comments since they present greater specificity, in terms of: a) the nature of the language, b) their conceptual grouping and c) their dynamic upgrading. Second, unlike other research works that consider only two social media platforms, Reddit and Twitter, our goal is to explore how different data sets and training models can be applied to more social media platforms.

This paper is a survey of social media summarization techniques and approaches as well as a presentation of our work toward a system that generates summaries of user comments on social media posts using transformers. The rest of the paper is organized as follows: Section 2 is a literature review of the research that has been carried out so far in the field of text summarization. Section 3 illustrates the importance of transformers against previous deep learning methods and gives a basic description of transformer model architecture. Section 4 depicts the motivation and design of our system and compares 3 pre-trained transformer models based on a selected dataset. Section 5 shows the results of the comparison between 3 pre-trained model pipelines and Section 6 contains concluding remarks and our future work.

2. Literature review

In the field of Automated Text Summarization on Social Media, several attempts have been made by researchers as seen in Table 1. APPENDIX I. We have grouped them in two main categories: 1) Pre-neural approaches that do not use neural networks and 2) Deep learning approaches that use neural networks. A further division of deep learning approaches has been done, based on transformer architecture, into the following three categories: 1) *Social media summarization systems that do not use Transformer architecture*, 2) *Transformer-based Social media summaries* and 3) *Transformer-based projects that do not generate social media summaries*. Finally, recent trends on large language models (LLMs) lead us to prompt engineering that give a new perspective to the fields of text generation and summarization. A more detailed presentation follows:

2.1 Pre-neural approaches

In the early stages of natural language research, summarizing web posts, microblogs, and social networks researchers used probabilistic and optimization methods to generate a summary from different types of data such as text, videos, images, hashtags and so on. In some cases the summary is not only text but a multimedia representation.

A typical example of multimedia representation is the model proposed by [10] & [11]. This model takes as input real-time user interaction in the social media stream and provides a multimedia representation as output. The summarization process takes minimal linguistic information into account. The output is a multimedia representation combining photos, video and some text. The algorithm uses an optimization process under a score-based input to find the topic of greatest importance. Specifically, for each Multimedia Social Network (MuSN), they use graph-based modeling and influence analysis methodologies to identify the most important media objects related to one or more topics of interest. Then a media summary is extracted from a list of candidate objects. This is achieved with the help of heuristic models based on the following properties: Priority, Continuity, Variety and Non-Repeatability. This project does not follow one of the text summary models, either extractive or abstract, because it does not create a text summary.

A second example of multimedia summarization is the model proposed by [12] which is a framework for generating a visualized summary from microblogs with multiple media types. The main goal of this work is to separate the different parts of an event published on social media by creating sub-events. Then for each secondary event it creates a clear and precise summary. Thus a fragmentary summary is produced. Each microblog is taken as a separate information mining unit, so a microblog digest is considered a multi-document digest (MDS). MDS faces two challenges: 1) Word limit and 2) Unstructured social media content. The proposed framework consists of 3 stages: 1) Noise removal in which the authors apply a filtering model to clean data that calculates the probability that the data is irrelevant. 2) Sub-event extraction where the authors propose a multi-media probabilistic model, called Cross-Media-LDA (CMLDA), which explores the associations between the different media types of the published event or separates the sub-events. 3) Creation summary. An algorithm is implemented to identify representative microblog

texts considering three criteria: vagueness, importance and diversity. Based on these criteria, the summarizing ability is also measured. The important point of this work is that the knowledge acquired in the previous stages is used, thereby reinforcing each other for the proper functioning of both the textual and visual summarizing mechanisms.

Another research work that considers each post as a separate text is the work of [13] & [14]. This model generates a summary using Twitter posts by performing a search on the Twitter API based on trending topics. It automatically generates an abstract based on multiple papers published on the same topic by applying a Phrase Reinforcement (PR) algorithm by adding the TF-IDF technique to it. The algorithm performs the following steps: 1) finds a specific phrase of each document that matches the topic. This is done on the one hand because users often use the same word or phrase to describe an event that usually matches the title of the event and on the other hand because very often the most relevant post is "retweeted". 2) Based on this phrase it starts asking Twitter.com to retrieve related posts, 3) filters posts based on related content. The longest sentence from each post is isolated. The collected sentences become the input to the PR algorithm, 4) creates a graph with a common root node representing the topic sentence. Next to the root node there are other nodes, the word nodes and the word count nodes that show how many times the word appears in the sentence, 5) the PR algorithm searches for the path with the highest word frequency (count) starting from the root node to the final. Following this path the words are combined and an initial summary is produced. The algorithm iterates using the partial summary each time as the root node and reconstructs the graph, arriving at the final summary.

In the work [15] the authors create a model that aims to summarize tweets that have sports as their topic. The key point in their research is finding real-time events, which is of particular importance in repetitive and real-time events such as sports. The design of the model includes two stages: 1) Application of the modified Hidden Markov Model according to which the timeline of events is segmented, based on the tweet-stream and the distribution of the words used in the tweets. Each part is a separate "sub-event" of different importance. 2) Selection of tweets that can provide information about the section that was considered most important.

A framework that searches for categories of topics on Twitter and then generates summaries of tweets corresponding to that topic is presented in [16]. To determine the topic categories to use in summarization, the time sequence of the tweet event is taken into account. An important factor is the detection of tweets written at the same time for an underlying event to extract important data. To this end they apply two topic models: the Decomposition Topic Model (DTM) and the Gaussian Topic Model (GDTM) that exploit the temporal correlation between tweets under predicted conditions to produce a concise and information-rich summary of events.

2.2 Deep Learning Models

Since deep learning algorithms require a ton of data to learn from, this increase in data generation is one reason deep learning capabilities have grown in recent years. Additionally, deep learning algorithms benefit from the most powerful computing

power available today, as well as the proliferation of Artificial Intelligence (AI) as a Service. Deep Learning has achieved significant success in various fields, such as computer vision and natural language processing. In the following paragraphs we present models that make use of neural networks to generate summaries. Various deep learning models have been employed, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and RNNs with Long-Short Term Memory (LSTM). In addition, recent years have seen the emergence of new techniques in abstractive summarization, such as those based on transformers. The specificity of these techniques lies in the use of an attention mechanism and not a recurrent or convolutional neural network.

The idea of transformers was first introduced in the paper "Attention is All You Need" [5] where transformers achieved state-of-the-art results for translation tasks over previous NLP models. The idea has been extended to all NLP tasks and many researchers have adopted transformer models to build their own applications. The social media summarization field is in the early stages of research. Referring to the literature we distinguish, so far, the following aspects:

- *Social media summarization systems that do not use Transformer architecture.*

The authors of the paper [17] introduce a project which summarizes social media posts by incorporating user comments in order to capture the reader focus. They use a model named reader-aware summary generator (RASG). Their model is based on a sequence to sequence framework rather than transformer architecture to generate abstractive summaries. It consists of 4 main parts: 1) Sequence-to-sequence summary generator, 2) an attention module which captures the reader's focus, 3) a supervisor module that matches the semantic gap between the generated summary and the reader's focus and 4) a goal tracker that defines the goal of each generation step. Specifically, the seq2seq architecture with an attention mechanism is used to generate summaries. First, words from user comments are correlated with those in the document. This correlation is then calculated because it is considered an important indicator of the "reader-centered aspect". At the decoding layer, attention weights are defined as the "decoder focused aspect" of the produced summary. Finally, a supervisor calculates the deviation of the "reader-centered view" and "decoder-centered view" with the goal of reducing the deviation.

Another attempt of text summarization that is related to social media posts and customer reviews is presented in [18]. The main idea is the use of a Sequence-To-Sequence Model with Attention approach to produce abstractive text summarization with no use of transformers. It is based on bidirectional RNN with attention layer on input data. The model uses also LSTM as encoder and decoder to build summaries. LSTM encoders partly extract a fixed length of sequence of the input and LSTM decoder generates a translation of this sequence. With the attention layer in this architecture the decoder is allowed to have more direct access to the input sequence.

The authors of [19] proposed a summary-aware attention model that aims to generate abstractive text summarization in social media. This mechanism helps the decoder to make the right decision. The authors have based this novel proposal on two

important limitations that have been faced on previous models: 1) the original text is not structured, like social network texts, making the decoder's task difficult, and 2) the calculation of attention weight does not take into account the summary information created in previous layers, which negatively affects the consistency of the summary. To overcome the above limitations, a new model of attention computation is proposed, called "summary-aware attention" mechanism for abstractive summary generation. According to this, attention is calculated at each level, taking into account the data of the previous level.

The work of [20] generates an abstractive text summary on social media, in Turkish. The data is collected and the Word2Vec model is used to preserve the meaning of the text, which is a very important factor for generating summaries. The GRU (Gated Recurrent Unit) neural network is then applied to divide user posts into positive and negative. Finally, using the Latent Semantic Analysis (LSA) method, a summary text in Turkish is generated from user comments on Twitter. The LSA algorithm accepts as input the content of a text and reveals the hidden semantic relationship between terms and sentences of that text.

The model of [21] produces an abstract digest of social media text based on Attentional Encoder-Decoder recurrent neural networks. The model is based on the encoder-decoder architecture by enhancing it with an additional hidden layer before the encoder unit. In this layer it is decided which information is useful and which should be removed. In this way, invalid information is better filtered in the encoder unit and through the reinforcement learning policy, better results are achieved. More specifically, this work meets the following three main contributions: (i) They have applied to summarization, the attentional RNN encoder-decoder originally developed for machine translation, showing that it already outperforms technology systems in two different English corpora. (ii) Since the model that is already used for machine translation addresses specific problems when applied in summarization model, the authors propose new models and show how additional performance is improved. (iii) Finally, they introduce a new dataset for the multi-sentence abstract summation task and establish benchmarks. The authors also propose a variety of selective gating methods to better filter the information while preserving the meaning of the original text.

- *Transformer-based Social media summaries.*

An abstractive Event Summarization on Twitter [8] framework uses a pre-trained BERT-based model as an encoder and a not pre-trained transformer model as a decoder. An event topic prediction component helps the decoder to focus on more specific aspects of posts. The most liked comments are used to produce more coherent summaries. The framework consists of the following features: 1) a mechanism that selects the most important tweets to be used by the decoder that is implemented in the following steps: a) tweets are selected based on their timestamp in ascending order. b) After the initial selection, in order to determine their importance, each tweet will be compared to each of those already selected. Selection is based on how representative and informative a tweet is. This gives the summary precision and coherence. 2) a mechanism that predicts the thematic category of the event. Topic information is used by the decoder to generate different summary styles from different events, thereby focusing on very specific aspects of topics. 3) A BERT model as encoder,

to better exploit the grammatical and semantic information of tokens pre-trained model and Transformer-based model as decoder. 4) two separate optimizers to smoothly integrate the pre-trained BERT model and the untrained transformer.

A new model is introduced in [22] which generates summaries of each text posted on a Chinese web platform. The model combines BERT, reinforcement learning and other technologies. Their work based on LCSTS, a dataset constructed from a Chinese social media platform named "Sina Weibo".

What is highlighted in [9] is the importance of summarizing user feedback. In their paper, they propose a multi-document text summarization scenario that includes meaningful comment detection, extending previous single-document approaches. They use BERT as the encoder and the transformer-based model as the decoder and produce an abstract summary of the embedded news discussions. This architecture is extended with an attention encoding level that is fed with user preferences. Attention encoding focuses on comments with the highest social impact. Consequently, the model summarizes the most relevant aspects of a discussion gathered from a news article. To encourage the model to pay attention to important comments, they introduce a data-driven strategy that focuses the model on those comments. The key element of this research is that to summarize the discussion there is the approach of many works by many authors. This has the consequence that users are considered co-authors of the original news story, to which they can add new information. Comments are rated as important based on user preferences.

A Transformer-based abstractive summarization is presented in [7] which creates summaries in three languages from posts and comment pools. The datasets are from Reddit and Twitter. The authors fine-tune T5 and LongFormer, test them against BART, and consecutively test them to pools of comments. In addition, they apply enlarged Transformer-based models on Twitter data in three languages (English, German, and French) to discover the performance of these models on data with non-English text. The main part of this work is the comparison of open source models for social media summarization. Three stages are applied: 1) Performance is optimized and one of three models is selected: LongFormer, Bart and T5. 2) The selected model is tested on pools of comments from Reddit and evaluated based on the similarity of posts and comment summaries. 3) The models are also applied to Twitter data in three languages (English, German and French). This study concludes that using transformer-based models, such as LongFormer and T5, give better results in generating summaries on social networks.

In [23] model for extractive summarization with transformers is introduced. This system summarizes a single web document by exploiting relevant information of social media to enhance the document summarization. The intuition behind this model is that they make use of relevant user posts and transformers to enrich the sentences of the output summary. They stack a Convolutional Neural Network (CNN) on the top of transformers (BERT) for classification.

- *Transformer-based projects that do not generate social media summaries.*

In [24] a two-stage transformer-based approach is proposed to generate abstractive summaries from articles in Chinese. The

model produces fluent and variable length abstractive summarization according to the user's demands. It consists of two modules: text segmentation module and two-stage transformer-based module. A pre-trained BERT model and a bidirectional LSTM are used first to divide the input text into segments. Subsequently the extractive based BERTSUM model is constructed in order to extract the most important information of the segments. Then, collaborative training is used to train the two-stage Transformer-based model. The training process begins with the document Transformer in the second stage. The input of the document Transformer is the outputs of the extractive model and the output of the document Transformer is the headline summary.

An abstractive summarization system of meeting conversations using transformers is presented in [25]. The system takes as input human dialogues and by applying the classic Transformer model summarizes the dialogues.

In their work [26] compare three pre-trained transformer-based models in an attempt to produce abstractive summarization of news articles on web. The pre-trained models used in this project are BART, PEGASUS and T5. When the transformer-based pre-trained language models are fine-tuned they give satisfactory results and fluent summaries. After evaluating each model with ROUGE the T5 model outperformed all other models.

Authors of [27] use dataset from Wikihow knowledge base and then deploy bidirectional encoder representations from transformers (BERT) and text-to-text transfer transformer (T5) models in order to produce abstractive summaries of the articles and then use the ROUGE scores to assess the performance of the two models and compare them.

2.3 Prompt Engineering

In the field of natural language processing (NLP), Prompt Engineering has emerged as one of the most innovative and powerful techniques for improving the performance and adaptability of language models. This recent technique has its roots in the development of LLMs whose history dates back to the 1950s, but until the introduction of BERT and GPT models in 2018, LLMs had the main role in NLP tasks. These models still use complex algorithms and massive amounts of training data to understand natural language. Their ability to learn patterns and structures in language has proven invaluable in various NLP tasks. However, due to the significant time and computational resources required to train these models, researchers have turned to prompt engineering. By designing and configuring the appropriate prompts, which can be shaped to improve behaviour and results of these models to achieve specific tasks. These tasks involve extracting information, summarizing text, answering questions, generating code, and classifying text. The detailed reference to this technique is beyond the purposes of this work, as in the current phase of our project we will use transformer models as presented in the following paragraphs.

3. Transformer models and methodology

3.1 Transformers and Model Explanation

Transformer models are a new development in machine learning and neural networks based on transfer learning. The

Transformer model was first proposed in the paper “Attention Is All You Need.” [5]. These models can handle the context of a text exceptionally well forming a state-of-the-art architecture in various NLP tasks such as translation, text generation, summarization, question-answering, classification and sentiment analysis. Transformer models are significant because:

- They have given exceptional results in many natural language tasks that use sequential data.
- Unlike recurrent neural networks (RNNs), they support parallelization and, they have replaced repetition with attention. Therefore calculations can be done simultaneously.
- Their architecture allows for the creation of better-quality models because they can handle better long distance dependencies. Based entirely on the attention mechanism, they solve the problem of Recurrent Neural Networks with Long-Short Term Memory (LSTM) and CNNs, which is mainly the inability to model larger sequences without data loss.

The architecture of the transformer model inspires from the attention mechanism used in the encoder-decoder architecture in RNNs to handle sequence-to-sequence (seq2seq) tasks. Unlike RNNs the transformer eliminates the factor of sequentiality meaning that, it does not process data in sequence which allows for more parallelization and reduces training time. The transformer works as encoder-decoder architecture because it is composed of two large building blocks: an encoder and a decoder. As we can see in (Figure1) the encoder is on the left and the decoder is on the right. Both encoder and decoder can consist of a stack of Nx identical layers (in original paper Nx = 6). Since the model does not contain any recurrence or convolution, it adds a positional encoding layer at the bottom of the encoder and decoder stacks to take advantage of the order of the sequence. Each layer mainly consists of Multi-Head Attention and feed forward layers. Inputs and outputs are first integrated in an n-dimensional space [5].

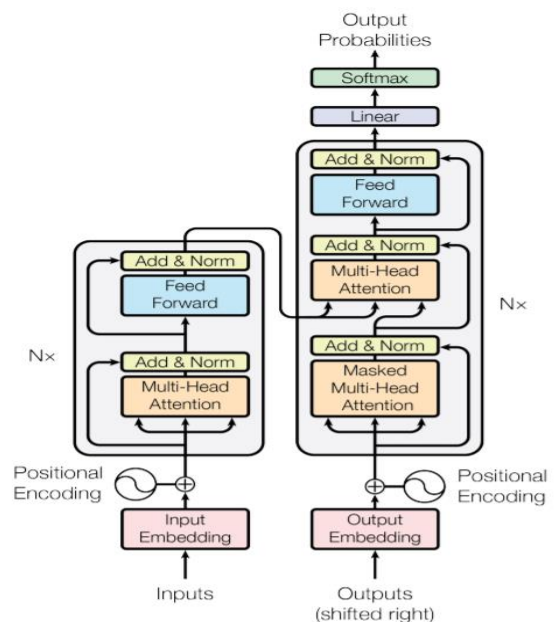


Figure. 1: Transformer Model Architecture

Encoder: The encoder is comprised of two major components: a multi-head attention (self-attention) mechanism which is followed by normalization and a feed-forward neural network.

- Inputs: Since transformers, as any other model, do not understand natural language, the input text is tokenized and processed to convert every word into a unique numeric id.
- Embedding layer: In the embedding layer the transformer uses learned embeddings to transform the input tokens into vectors of dimension $d = 512$.
- Positional encoding: A positional encoding is a fixed-size vector representation that provides the transformer model with information about where the words are in the input sequence, since there is no use of any recurrence or convolution.
- Multi-head Attention and Self-Attention: Self-attention is a mechanism that captures the contextual relationship between the words of a sentence. Based on a scale dot-product attention, it creates a vector of every input word. The main role of this vector is to understand how relevant every word in the input sentence is, with respect to other words in the sentence. Since self-attention is not only applied once, but also several times (in the original paper it is applied 8 times), is called multi-head attention. The objective is to generate several attention-based vectors for the same word. This helps the model to have different representations of the words' relations in a sentence. The different attention-based matrices generated by the multiple heads are summed and passed through a linear layer to reduce the size to a single matrix.

Decoder: The decoder side has a lot of shared components with the encoder side. The decoder takes in two inputs: (a) the output of the encoder and the output text shifted to the right. Then it applies multi-head attention twice with one of them being "masked". The final linear layer in the decoder has the same size of words as the target dictionary. Each unit will be assigned a score. The softmax function is applied to convert the scores into probabilities indicating the probability of each word to be present in the output.

4 Systems design

4.1 Motivation

Automatic Text Summarization in both formal documents (such as articles and books) and Social Media texts are grouped into two main categories: Extractive and Abstractive summarization. Initially, researchers focused on creating summaries using the extractive summarization method. According to this technique, the summarizer extracts important

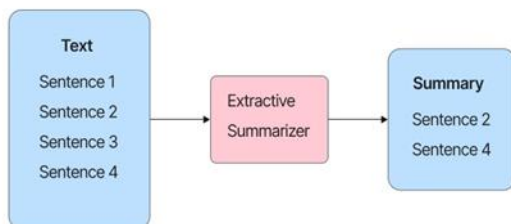


Figure 2. Extractive Text Summarization

words or sentences from the original document. In order not to lose its meaning it uses statistical and linguistic methods. It then rearranges slightly these words and phrase, to produce the output summary. This technique, while simple to apply, does not produce satisfactory results, as the summary is not coherent and accurate. This effect is enhanced when the length of the sentence starts to grow.

In recent years, researchers have turned to the abstractive summarization technique. In this technique the summarizer extracts the meaning of the original document and the output document is produced without losing this meaning. This process adopts the way the human mind produces summaries and despite its complexity, it tends to give better results. All of the research papers that applied the Abstractive Summarization technique used deep neural network methods and attention mechanisms to train their models, adopting the way the human mind might be trained to summarize documents.

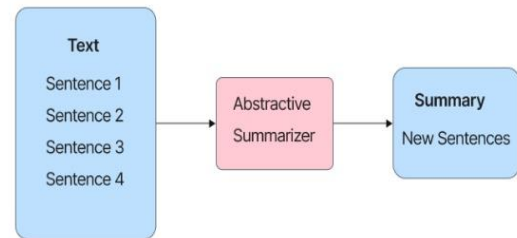


Figure 3. Abstractive Text Summarization

The area of social media summarization is in its early stages and remains challenging due to the particularity of social media content. As already mentioned in the literature review, various models, based either on pre-neural or deep learning techniques, have been proposed giving interesting results, but also facing several limitations. Transformer-based neural networks try to overcome these limitations giving state-of-the-art results. The perspective of our work concerns the following areas:

- Transformer-based models.
- Abstractive summarization techniques
- The discussions of users under the post

4.2 Methodology

As we have mentioned earlier Social Media posts and user comments provide great challenges on text summarization: 1) tend to be informal, consisting of slang expressions and special characters, 2) show deviations from the original theme and dependencies on previous opinions and, 3) since they are short, they lack lexical richness, 4) due to the massive amount of comments generated under each post, many of them are repetitive. It is therefore considered necessary to use data pre-processing techniques in order to improve the performance of the model. The original idea of our approach is to use the pre-trained transformer models. For a correct decision we fed with data 3 pre-trained pipelines of T5, BART and PEGASUS models comparing the generated summaries (Figure. 4). All three of the aforementioned pre-trained models generate abstract summaries and use the

encoder-decoder transformer architecture that performs best in text generation tasks. Our model uses a Facebook dataset of news posts and their corresponding comments and consists of the following steps:

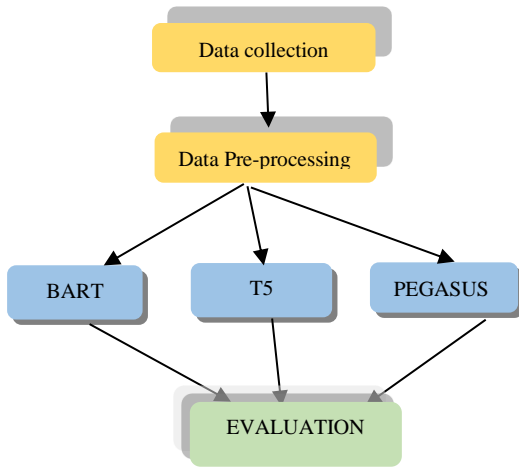


Figure 4. Pre-trained model evaluation

- Data collection and pre-processing using Python libraries.
- Feed this text into the transformer-based encoder.
- Clustering and attention rendering based on the topics.
- Feed this encoding to the Transformer decoder and
- Produce the summary text

4.3 Data Collection and pre-processing

Finding the right datasets from social media is a really complex task. Especially for data that will be used for summary. Several social media platforms are limited in receiving data, and in those that are open, the data must be reformatted to meet the needs of the project. The dataset we are looking at for this project is Facebook news posts accompanied by user comments below each post. The raw data has 7 columns, namely ‘created_time’, ‘from_id’, ‘from_name’, ‘message’, ‘post_name’, ‘post_title’, ‘post_num’ and 1,781,576 rows. The dataframe consists of the 3 columns: ‘message’, ‘post_title’, ‘post_num’ where the ‘post_title’ as well as the ‘post_num’ identify a specific post while the ‘message column represents the user comments under the post. We focused on creating summaries of user comments, as they are more discrete in terms of (a) the nature of the language and (b) their conceptual grouping. After downloading the dataset and reshaping it in a meaningful way for summary generation, the first step is to apply some basic pre-processing before building the model. Using impure and unclean data does not lead to issuing the desired results. Hence in this step, using python packages NLTK and regex, we have removed punctuation, special characters, emoji and link threads, as long as NULL values. The second step is to perform data clustering according to the topic of the discussion. Since we on the first stages of our ongoing research work we need to decide which of the pre-trained models can be better trained and fine-tuned based on our dataset. Therefore, the cleaned and classified data is first fed to the pre-trained-models and the results of the comparison are extracted.

4.4 Pre-trained Transformer models and pipelines

Today excellent results have been achieved in all tasks thanks to the wide variety of pre-trained transformer models available. Pre-trained language models have been trained on large-scale datasets and due to their capabilities and recourses can be reused even with small amount of datasets. PTLMs are differentiated based on their architectures and pre-training tasks. In order to make a correct choice of the appropriate model for a particular task on a particular dataset, knowledge of these details is crucial. For example, BERT is better suited for task understanding than task generation. Furthermore, another differentiation of pre-trained models is their architecture. There are many models whose architectural backbone is Transformers, but some of which pre-train only the encoder, such as BERT and UniLM, while others pre-train only the decoder, such as GPT. Therefore, PTLMs must be chosen wisely based on these details when adapting them to downstream tasks. Based on the above, in order to proceed with the correct model selection, we have set the following criteria:

- Both input and output of the system are text
- The output will be abstractive summary generation, which means that we chose the models that produce abstractive summaries and are based on the Encoder-Decoder architecture [28].
- All the pre-trained models can fine-tuned on our target task according to the limitations of our datasets.

The HuggingFace hub [29], acts as an open source library and provides a huge number of pre-trained models as well as datasets, on a wide range of different NLP tasks. These models can be used to predict a summary and then accurately fit any data set. Due to the large number of models available in Hugging Face, it is quite difficult to decide which one to use as the best one for our dataset in order to produce the desired results. Pipelines are a fast, easy and efficient way to use different pre-trained models and can be applied to many NLP tasks. The pipelines meeting the above criteria are BART, T5 and PEGASUS and are presented thoroughly below:

- BART – stands for Bidirectional and Auto-Regressive Transformers [30]. BART supports tasks involving Natural Language Creation, such as the summarizing task, which requires new text to be generated to summarize a large input text, unlike the traditional summarizing task that extracts some parts of the input text to make the summary. It combines the pre-training processes of BERT and GPT within the encoder-decoder architecture (Figure 5). Input sequences undergo one of many possible transformations,

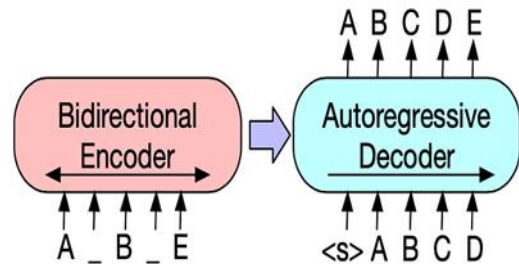


Figure 5. BART (source original paper)

from simple masking to model sentence permutation, token deletion, and document rotation. These modified inputs are passed through the encoder and the decoder must reconstruct the original texts. This makes the model more flexible, as it can be used for NLU as well as NLG tasks, and achieves top performance in both. BART pre-training involves 2 steps: a) the input text is free of unnecessary noise, which may change the length of the text. b) the seq2seq model learns to reconstruct the original text from the corrupted text. BART is known for its excellent performance in tasks that require complex language handling, such as text summarization and machine translation. BART's pre-training task encourages the model to learn representations that are robust to noise and variations in the input text. This makes BART suitable for tasks that require handling text that is noisy, ambiguous, or written in different languages. It has a more complex architecture, which makes it more suitable for tasks that require handling large sequences of text or that require text generation.

- T5 - stands for "Text-to-Text Transfer Transformer" [31]. The T5 model is a task-agnostic model meaning that unifies all NLU and NLG tasks by converting them into text-to-text tasks, such as Translation, Language Inference, Information extraction and Summarization. All tasks are framed as sequence-to-sequence tasks, where adopting encoder-decoder architecture is natural thus it's capable of

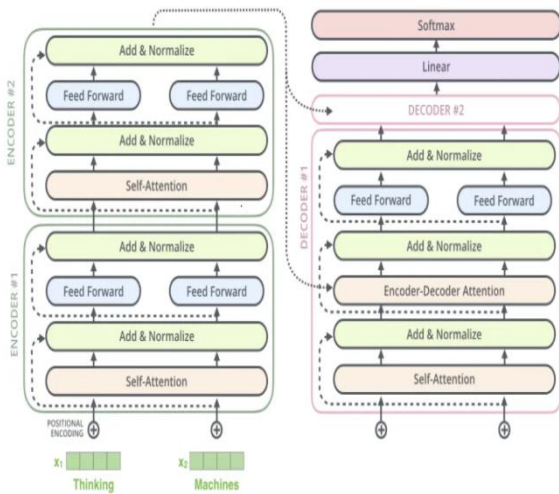


Figure. 6. T5 model (source original paper)

handling architecture uses the original Transformer architecture different sequence lengths for input and output. The T5 using the large crawled C4 dataset, the model is pre-trained with masked language modelling as well as the SuperGLUE tasks by translating all of them to text-to-text tasks. The largest model with 11 billion parameters yielded state-of-the-art results on 17 of the 24 tasks tested. It achieved high scores on Corpus of Linguistic Acceptability (CoLA), Recognizing Textual Entailment (RTE) and Natural Language Inference on WNLI tasks. It also achieved very high performance on the SQuAD dataset but less satisfactorily on the CNN/Daily Mail abstractive summarization task. The pre-training

process involves two types of training: supervised and self-supervised. During supervised training, downstream tasks from the GLUE and SuperGLUE benchmarks are used and tasks, as explained before. On the other hand, self-supervised training is done using corrupted tokens. This converted into text-to-text is achieved by randomly removing 15% of the tokens and replacing them with individual sentinel tokens. The encoder takes the corrupted sentence as input, while the decoder takes the original sentence as input. The target is then the dropped-out tokens, delimited by their sentinel tokens.

- PEGASUS – stands for Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence-to-sequence models [32]. It is also based on the Transformer architecture but with some modifications for abstractive text summarization. It is a task-specific architecture where every component in pre-training closely maps text summarization. PEGASUS uses a gap-sentence generation task, where whole encoder input sentences are replaced by a second mask token and fed to the decoder, but which has a causal mask to hide the future words like a regular auto-regressive transformer decoder. It is designed explicitly for abstractive text summarization, where it generates summaries that have the same meaning but different phrases from the source text. The model is pre-trained on CNN/DailyMail dataset, which consists of large volumes of articles. The pre-training objective, Gap Sentence Generation (GSG), is designed to have a well performance on text summarization and text generation tasks and is not used in other domains. It is adaptable and can be fine-tuned on limited data, achieving human performance.

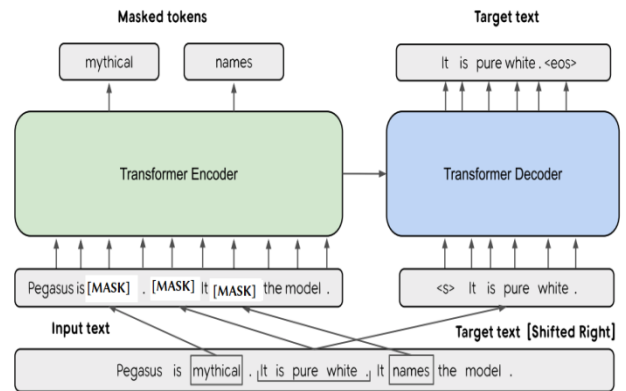


Figure. 7. PEGASUS model (source original paper)

5 Results

Measuring performance in a text generation task is not as easy as in other domains such as sentiment analysis or named entity recognition. It is very important to use the appropriate evaluation metrics to illustrate the best performance of the models. To measure the performance of our own work we used one of the most well-known NLP job evaluation mechanisms, ROUGE. ROUGE stands for "Recall-Oriented Understudy for Gisting Evaluation" and evaluates the similarity between a candidate document and a collection of reference documents. The use of the

ROUGE score to evaluate the quality of document translation and summarization models [33]. There are variations of ROUGE metrics such as ROUGE-N, ROUGE-L and ROUGE-Lsum where the N in ROUGE-N stands for the numbers 1 and 2. More specifically, ROUGE-1 compares the monograms between the machine-generated summaries and the human reference digest, ROUGE-2 uses digrams instead of monograms, while ROUGE-L does not compare n-grams, instead treating each digest as a sequence of words and then searching for the greatest common subsequence (LCS). These measures, by comparing the ground truth and the generated text, automatically determine the quality of a generated summary. The number of overlapping n-grams, word sequences, and word pairs between the generated text and the ground truth are used for the metrics. In this work we use ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-Lsum. In Table 2 we show the results of the comparison between 3 pre-trained model pipelines, T5, BART and PEGASUS, on a sample text of Facebook news posts and comments. As we have already mentioned above, the use of these particular three models was made because all of them meet the following constraints: they based on Transformers with the Encoder-Decoder architecture and they produce abstractive summarization. T5 pipelines give better results than BART and PEGASUS. We are led to the conclusion that by applying this model to our own data we will produce more accurate summaries. Moreover, T5 is a modern model that has already been successfully used in summarization studies, but in the field of social media it is less explored. This is an initial estimate for our own model design, but by enriching our study with additional assumptions to optimize the results the pre-trained models will be re-evaluated for their performance.

Table 2. Evaluation and Comparison of ROUGE scores

Models	Evaluation Metrics			
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-Lsum
T5	0.585034	0.524138	0.544218	0.585034
BART	0.529032	0.483660	0.516129	0.529032
PEGASUS	0.365079	0.306452	0.349206	0.349206

6 Conclusions and future work

The constant increase in the volume of information on social media makes the summarization of texts decisive and necessary to save users' time and resources. Social media is event based. When an event occurs, social networks are flooded with posts related to the event as well as user comments under each post. Since the volume of published data is growing dynamically, reading all these posts in a reasonable time is almost impossible for anyone. Summarizing your social media feed posts is important to keep your readers informed correctly, accurately and in a timely manner. On the other hand, another important issue is the summary of user comments and sub-comments under each post. As their number increases, reading them is a time-consuming and difficult process. User comments are important to read because they express the public's opinion on an important topic. For this reason creating a summary of the comments is of utmost importance. Social media is a challenging area because it offers different types of information, but also because online www.astesj.com

discussions and chat threads are not formal. They usually contain slang expressions, special symbols such as hashtags or emoticons. Generating summaries from user comments is a difficult task, since they confront language informality and lack lexical richness. Additionally, there is one main question: "How many and which comments will the summary model consider". Comments are updated dynamically. Working with the most liked, is not an option, as it has been observed that comments readers pay attention to, are the initial comments on the list, finding it tedious to go even deeper. In this paper the overall literature of social media text summarization was reviewed. The result of this literature review shows that various technologies can be used to develop social media summarization models. However, each technology is implemented under different assumptions and constraints. Research around social media text summarization and, in particular, those involving abstractive summarization, are based mainly on Neural Network models. In light of these models, our approach focuses on developing an abstractive, Transformer-based summarization system. The field of text summarization in social media, especially with Transformer models, is still underexplored for both posts and user comments. We focused on creating summaries of user comments since they present greater specificity both in terms of the nature of the language and in terms of their conceptual grouping. In particular, we took user comments on facebook posts and after pre-processing we used them as input to pre-trained language models, based on transformer encoder-decoder architecture. We used ROUGE metrics to compare 3 different pre-trained models and we concluded that T5 gives the highest performance, leading us to the conclusion that T5 will be applied better to our dataset. We proceeded to cluster the data so that T5 can be fed and fine-tuned with the desired text to be summarized. Certainly this is an initial consideration in the design of our model. Through our research we learned about several state-of-the-art models that use a transformer architecture for text summarization. Despite adopting the transformer architecture, these models differ in pre-training strategies and results. Choosing the most appropriate model is necessary, considering the size and training data set of the model. For future work we consider optimizing the results and generating summaries by fine-tuning or prompt-tuning the models to achieve even better outputs. Collecting and classifying datasets from various social media platforms is a difficult task thus our intention is to check the performance of multiple datasets. Since this is an on-going research process we will be able to present more detailed results shortly.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The publication fees were totally covered by ELKE at the University of West Attica.

References

- [1] A. Nenkova, K. McKeown, "A survey of text summarization techniques", Springer US: 43–76, 2012, doi:10.1007/978-1-4614-3223-4_3.
- [2] V. Varma, L.J. Kurisinkel, P. Radhakrishnan, "Social Media Summarization", In A practical Guide to Sentiment Analysis, Chapter 7 pp.135-153. 2017.

- [3] D. Suleiman, A. Awajan, "Deep Learning Based Abstractive Text Summarization: Approaches, Datasets, Evaluation Measures, and Challenges," *Mathematical Problems in Engineering*, **2020**, 2020, doi:10.1155/2020/9365340.
- [4] V. Gupta, G.S. Lehal, "A Survey of Text Summarization Extractive techniques," in *Journal of Emerging Technologies in Web Intelligence*, 258–268, 2010, doi:10.4304/jetwi.2.3.258-268.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, "Attention Is All You Need," In 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA., June 2017.
- [6] S. Haque, Z. Eberhart, A. Bansal, C. McMillan, "Semantic Similarity Metrics for Evaluating Source Code Summarization," in *IEEE International Conference on Program Comprehension*, IEEE Computer Society: 36–47, 2022.
- [7] I.S. Blekanov, N. Tarasov, S.S. Bodrunova, "Transformer-Based Abstractive Summarization for Reddit and Twitter: Single Posts vs. Comment Pools in Three Languages," *Future Internet*, **14**(3), 2022, doi:10.3390/fi14030069.
- [8] Q. Li, Q. Zhang, "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, 2020, doi:10.1145/3366424.3382678.
- [9] I.T. Palma, M. Mendoza, E. Milios, "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 2021.
- [10] 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)*.
- [11] F. Amato, A. Castiglione, F. Mercorio, M. Mezzanica, V. Moscato, A. Picariello, G. Sperli, "Multimedia story creation on social networks," *Future Generation Computer Systems*, **86**, 412–420, 2018, doi:10.1016/j.future.2018.04.006.
- [12] 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, doi:10.1109/TMM.2014.2384912.
- [13] J. Kalita, B. Sharifi, M.-A. Hutton, "Summarizing Microblogs Automatically". *Association for Computational Linguistics, Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 685–688, 2010.
- [14] B. Sharifi, D. Inouye, J.K. Kalita, "Summarization of Twitter Microblogs". *The Computer Journal*, Volume 57, Issue 3, March 2014, Pages 378–402, <https://doi.org/10.1093/comjnl/bxt109>
- [15] 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
- [16] F. Chong, T. Chua, S. Asur, "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>, 2021.
- [17] S. Gao, X. Chen, P. Li, Z. Ren, L. Bing, D. Zhao, R. Yan, "Abstractive Text Summarization by Incorporating Reader Comments". In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.
- [18] P. Bhandarkar, K.T. Thomas, "Text Summarization Using Combination of Sequence-To-Sequence Model with Attention Approach", Springer Science and Business Media Deutschland GmbH: 283–293, 2023, doi:10.1007/978-981-19-3035-5_22.
- [19] Q. Wang, J. Ren, "Summary-aware attention for social media short text abstractive summarization," *Neurocomputing*, **425**, 290–299, 2021, doi:10.1016/j.neucom.2020.04.136.
- [20] A. Varol, "Innovative Technologies for Digital Transformation". In the 1st International Informatics and Software Engineering Conference (IISEC-2019) proceedings book : 6-7 November 2019, Ankara/Turkey.
- [21] Z. Liang, J. Du, C. Li, "Abstractive social media text summarization using selective reinforced Seq2Seq attention model," *Neurocomputing*, **410**, 432–440, 2020, doi:10.1016/j.neucom.2020.04.137.
- [22] Z. Kerui, H. Haichao, L. Yuxia, "Automatic text summarization on social media," in *ACM International Conference Proceeding Series, Association for Computing Machinery*, 2020, doi:10.1145/3440084.3441182.
- [23] 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, doi:10.1109/KSE50997.2020.9287388.
- [24] 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, doi:10.1109/TASLP.2020.3006731.
- [25] 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, doi:10.1109/INOCON50539.2020.9298305.
- [26] A. Gupta, D. Chugh, R. Katarya, "Automated News Summarization Using Transformers.", In *Sustainable Advanced Computing*, 2022, Volume 840. ISBN : 978-981-16-9011-2, 2022
- [27] A. Pal, L. Fan, V. Igodifo, Text Summarization using BERT and T5. https://anjali001.github.io/Project_Report.pdf
- [28] 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, doi:10.1109/SMART50582.2020.9337081.
- [29] 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.
- [30] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension," 2019.
- [31] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," 2019.
- [32] J. Zhang, Y. Zhao, M. Saleh, P.J. Liu, "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization," 2019.
- [33] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries". In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics, 2004

APPENDIX I

Table 1: Methods and Summarization Techniques on Social Media

Author	Summarization Technique	Implementation Method	Pre-neural Approaches	Deep-Learning Approaches		
				RNN/CNN	Transformer-based	Social Media
[10]	Summarization Graph with multimedia output	Influence Maximization and ABC algorithms	✓	✗	✗	✓
[11]	Summarization Graph with multimedia output	Influence Maximization and ABC algorithms	✓	✗	✗	✓
[12]	Extractive Summarization	A cross-media probabilistic model, termed Cross-Media-LDA (CMLDA),	✓	✗	✗	✓
[13] & [14]	Extractive Summarization	Phrase Reinforcement Algorithm & TF-IDF algorithm	✓	✗	✗	✓
[15]	Extractive Summarization	A modified Hidden Markov Model	✓	✗	✗	✓
[16]	Extractive Summarization	Gaussian Decay Topic Model (GDTM)	✓	✗	✗	✓
[17]	Abstractive Summarization	Reader-aware model based on CNN and Bi-RNN with Seq2Seq (No Transformer)	✗	✓	✗	✓
[18]	Abstractive Summarization	Sequence-to-Sequence with Attention Approach (No Transformer)	✗	✓	✗	✓
[21]	Abstractive Summarization	Selective Reinforced Sequence-To-Sequence Attention Model (No Transformer)	✗	✓	✗	✓
[19]	Abstractive Summarization	Summary-aware attention neural model (No Transformer)	✗	✓	✗	✓
[20]	Abstractive Summarization	Word2Vec with Recurrent Neural Network model GRU (No Transformer)	✗	✓	✗	✓
[8]	Abstractive Summarization	NNs with BERT model as the encoder and Transformer architecture as the decoder	✗	✗	✓	✓
[22]	Extractive Summarization	BERT, reinforcement learning and other technologies	✗	✗	✓	✓
[9]	Abstractive Summarization	NNs with BERT model as the encoder and Transformer architecture as the decoder	✗	✗	✓	✓
[7]	Abstractive Summarization	Transformer-based NNS LongFormer Large and T5 Large	✗	✗	✓	✓
[23]	Extractive Summarization	BERT with use of CNN	✗	✗	✓	✓
[24]	Extractive Summarization	Pre-trained BERT, RNN/LSTM, BERTSUM	✗	✗	✓	✗
[25]	Abstractive Summarization	Classic Transformer Model	✗	✗	✓	✗
[26]	Abstractive Summarization	Pre-trained Transformer Models BART, PEGASUS, T5	✗	✗	✓	✗
[27]	Abstractive Summarization	BERT, T5	✗	✗	✓	✗